Trajectory estimation using an iterative method

TREBALL FINAL DE GRAU

Treball realitzat per:
Adrià Torres Ventura

Dirigit per:
Prof. Wen-long Jin(University of California, Irvine)
Prof. Francesc Soriguera(UPC)

Grau en:
Enginyeria Civil

Barcelona, 22 de juny de 2016

Departament d'Infraestructura del Transport i del Territori

Abstract

In a future scenario where autonomous vehicles will coexist with the traditional cars , the trajectory can contribute with an important role. Basically, knowing and predicting the trajectory of the other vehicles can lead to a better autonomous trajectory in terms of travel time, security and energy efficiency. This project upgrades a theoretical trajectory estimation method in order to improve his accuracy through an iterative method between trajectory and traffic estimation. Besides, this method follows the aim to estimate the trajectory without previous fundamental variables available, just with the information provided by Eulerian and Lagrangian data. This data is provided by the Loop detectors and the vehicle reidentification respectively. To do that, we take into account the FIFO violation trajectory estimation method developed in (Rey and Jin, 2016) and the traffic estimation method by (Zhe Sun, 2016). In the last one, we make some modifications in order to be available for the overtaking scenario where FIFO is violated. With the NGSIM datasets, we demonstrate that the iterative method reaches great results in terms of estimation, as the model tends to reach a convergence for both trajectory and parameters estimation. To get these better results is demonstrated that a discretization method is recommended in the parameters estimation procedure. Finally, an Autonomous Vehicles(AV's) implementation is considered and analysed in terms of properties, which affects directly into a more accurate definition of the order-change function.

Abstract

[Versió Català]

En un futur pròxim els vehicles autònoms guanyaran un pes molt important dins l'automobilisme. En aquest nou panorama, conèixer i poder estimar amb precisió les trajectòries pot contribuir en una millor configuració tant dels vehicles autònoms com de la xarxa en termes de seguretat, temps de viatge i seguretat. Aquesta millora es deu al control remot que es tindrà sobre aquests nous vehicles, els quals podrem imposar la trajectòria més òptima en funció de l'estat real del trànsit. És aquí doncs on radica la importància del coneixement d'aquestes. En aquest projecte es busca millorar l'estimació de les trajectòries a partir d'un mètode iteratiu entre una estimació teòrica a partir de l'equació de Newell i l'estimació dels paràmetres fonamentals del trànsit que es veuen involucrats.A més, la utilització d'aquesta mètode no requereix aquestes paràmetres com a condicions inicials, com passava en antics estudis, sinó que només necessitem una suposició inicial. Cal destacar però que el model requereix la informació proporcionada pels bucles d'inducció i el sistema d'identificació dels vehicles en l'entrada i sortida del segment estudiat. Per dur a terme aquest treball considerem el mètode d'estimació de les trajectòries considerant avançaments desenvolupat per (Rey and Jin, 2016) i l'estimació de paràmetres tractada per (Zhe Sun, 2016). En aquest últim s'han afegit certes modificacions per tal de fer compatible el model amb l'escenari d'avançaments proposat. Posteriorment, amb la base de dades proporcionada pel NGSIM hem sigut capaços de testar i demostrar els bons resultats que produeix el model iteratiu el qual tendeix cap a un valor de convergència, tant per la trajectòria estimada i l'error com pels paràmetres. Finalment, es considera la introducció del vehicle autònom en l'estudi i s'analitza quines en són les seves conseqüències. A partir d'una sèrie d'hipòtesis s'arriba a una formulació més acurada de la funció d'ordre que defineix el comportament d'avançament dels vehicles.

1 Introduction

The chance to have information about the vehicle's trajectories can contribute enormously to benefit the transportation systems and the use of its users. The way to obtain this information varies depending on how are your available resources. Specifically, by estimating the trajectories you are able to obtain a real time status of the road conditions which can be really useful. The mainly applications are related with the better knowledge of the acceleration rates and traffic parameters of the road studied.Besides, a recent application which can lead to a huge benefits is related with the Autonomous Vehicles (AV'S). By knowing the real traffic status the AV's can establish and adapt his own trajectory to the one that better fits the environment in terms of travel time, security and energy efficiency. The estimation of trajectories has been studied in many different ways depending on the use of the technologies. Video-cameras processing images, global-positioning-system(GPS) data and even geographic-information-system(GIS) have been used in this direction, each one offering a method to estimate the trajectories with each own accuracy as is the case of (Barrios and Motai, 2011). Even though, this type of studies have a huge limitation as the use of this new and expensive technology cannot be installed everywhere. However, other studies with more theoretical vision have been done regarding the traffic flow theory. In (Coifman, 2002) a first estimation of both travel time and trajectory is studied using dual loop detectors. The huge disadvantage that this and similar works had is the FIFO(First-In-First-Out) assumption, which results in not being real and makes the estimation inaccurate. In both (Jin et al., 2006) and (Jin and Li, 2007) is demonstrated that the FIFO supposition is usually transgressed on multi-road highways. Taking into account this lack in the theoretical side, a recent work done by (Rey and Jin, 2016) has developed a trajectory estimation method with FIFO violation. This particular method is based on the Newell's simplified kinematic wave model (Newell, 1993a) which obtains individual vehicle trajectories from both Eulerian data(provided by the count made by the loop detectors) and the Lagrangian data(provided by the reidentification systems and GPS). In order to perform the FIFO violation scenario a linear order-changing model is assumed, which has the aim to represent the overtaking process.

In the present study we follow the aim to upgrade this method into an iterative scenario in order to both improve the accuracy of the procedure and avoid knowing the necessary

parameters in the Newell's model (Newell, 1993a). This parameters are from the fundamental triangular diagram and the initial conditions of the road.
The iterative method developed is between the trajectory estimation mentioned and the traffic parameters estimation method introduced in (Zhe Sun, 2016). This method is based in an optimization problem through the least squares approach. However, as we said there is a gap in the FIFO violation assumption which is also present in the traffic parameters estimation. Therefore, the parameters estimation is developed in the FIFO violation scenario so as to fit in the iterative process. Besides, a tolerances ratios are defined relative to the convergence the process reaches.
Moreover, as the AV's are the near future of the motoring world, we introduce an analysis of how the implementation can contribute in a better trajectory estimation. Specifically, we study how the order-change function will vary taking into account several assumptions related with the AV's.
Finally, we also test all the methods developed using the Next Generation Simulation(NGSIM) data (USDOT, 2008). In this stage, we need to make some treatment and considerations in the data available as this data is not provided by the source we assume in the method(Loop detectors and vehicle reidentification). Besides, we propose a vehicle discretization approach in the iterative method in order to maximize the accuracy of the results.

A list of the notation used is provided in the following table:

Table 1: Table of Notations

$F(t)$	The observed cumulative count at the upstream from 0 to t
$G(t)$	The observed cumulative count at the downstream from 0 to t
n_0	The initial number of vehicles within the segment studied
$n(x,t)$	The cumulative count at location x from 0 to t
V	Free-flow speed
W	Shock-wave speed
K	Jammed density
$X_i(t)$	Location of vehicle i at time t
Δt	Time step size (In this case 1/10 seconds)
$\theta(t)$	Order-change function
l	Length of the road segment
r_i	Entry time of vehicle i
s_i	Exit time of vehicle i

Contents

List of Tables

List of Figures

2 Literature Review of Trajectory Estimation

This study follows the estimation trajectory method developed in Rey and Jin (2016). In this particular case we focus in the FIFO violation case due to his greatest results and accuracy demonstrated. First, a summary of the fundamental theory used is provided.

2.1 Newell's Simplified Kinematic Wave Model

It is well known that one of the most used models to describe the traffic flow is the Lighthill-Whitham-Richard(LWR) kinematic wave model (Lighthill and Whitham, 1955). The model is based on the mass/traffic conservation which states the equilibrium between the vehicles flux through a road segment and the variation of the number of vehicles. This is described by a first order partial differential equation:

$$\frac{\partial k(x,t)}{\partial t} + \frac{\partial \phi(x,t,k(x,t))}{\partial x} = 0, \tag{1}$$

where $q(x,t) = \phi(x,t,k(x,t))$ is the time- and location-dependent fundamental diagram (Greenshields, 1935) and $k(x,t)$ is traffic density. This equilibrium equation can be understood through the following figure where a section with several trajectories are represented. The colourful boundaries, green and yellow, indicates the variation in this section of k and q respectively.

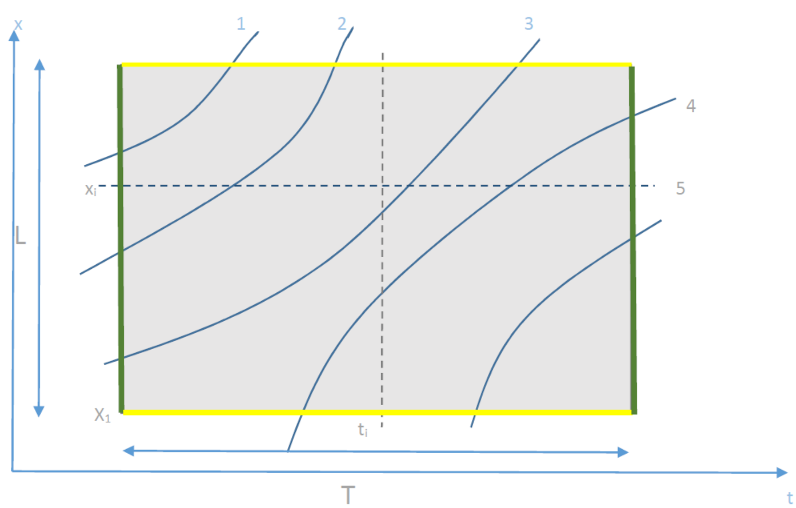


Figure 1: Representation of the LWR equilibrium equation

Later, in (Daganzo, 1997; Newell, 1993a) the q-k relation was upgraded to the most recognized and used fundamental triangular flow-density relation:

$$\phi(k) = \min\left\{Vk, W \cdot (K - k)\right\} \tag{2}$$

where V is the free-flow speed, W the shockwave speed in congested traffic, and K the jam density.

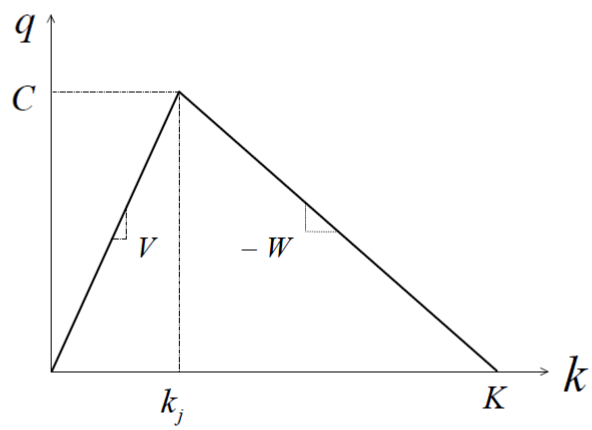


Figure 2: Fundamental triangular flow-density relation

Further in Newell's model (Newell, 1993a) it was shown that the kinematic wave model could be faced using a cumulative flow $n(x,t)$ as a state variable.The model describes, in a simpler way, the traffic conditions on a homogeneous road segment of length l from $x = 0$ to $x = l$ using the cumulative flows, $n(x,t)$. In the following figure is shown how the function is made up. This particular case is done from the section x_1 shown in the figure1.

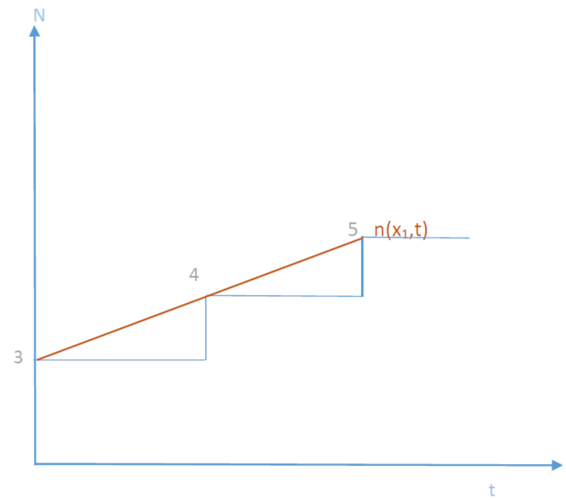


Figure 3: Representation of the function $n(x,t)$

So, once we have define n(x,t) as a continuous function we are able to state the flow-rate as:

$$q(x,t) = \frac{\partial n(x,t)}{\partial t}, \tag{3}$$

and the density as

$$k(x,t) = -\frac{\partial n(x,t)}{\partial x}. \tag{4}$$

At this point we define $G(t)$ and $F(t)$ as the corresponding observed cumulative flow at the downstream and upstream boundary from 0 to t. In the upstream case it has to be taken into account the initial number of vehicles within the segment, n_o. These vehicles has entered into the segment before t=0 and they are not been counted by the function $F()$. As a result, n_o is a parameter which has to be estimated.
Thus, the downstream count is,

$$G(t) = n(l,t). \tag{5}$$

and the upstream count,

$$F(t) + n_0 = n(0,t). \tag{6}$$

Taking into account this definitions we can use the Newell's simplified kinematic wave model (Newell, 1993a), where the cumulative flow at any location on the road $x \in [0,l]$ and time

$t > 0$ can be calculated using the following variational principle (Daganzo, 2005):

$$n(x,t) = \min\left\{F(t - \frac{x}{V}) + n_0, G(t - \frac{l-x}{W}) + (l-x)K\right\}. \tag{7}$$

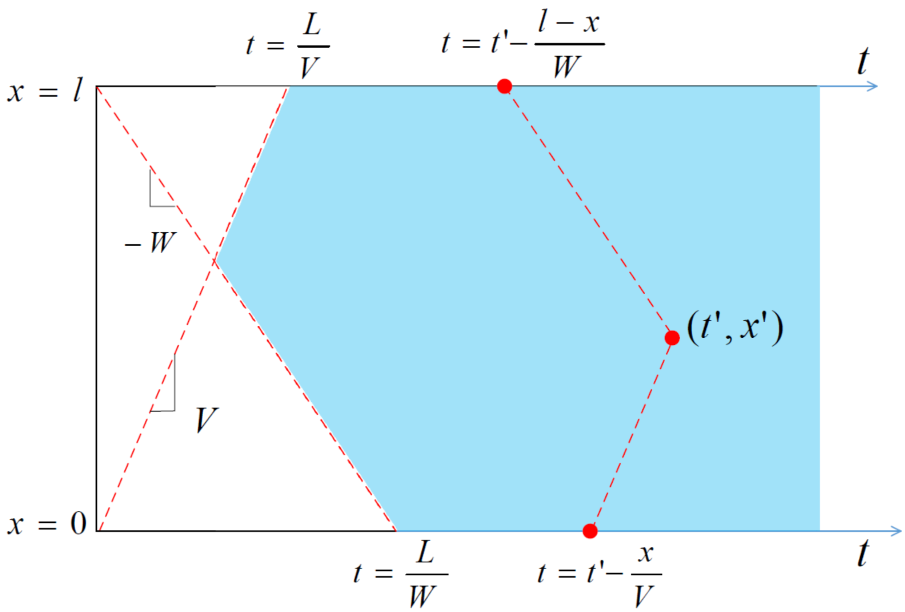


Figure 4: Representation of the Newell's simplified kinematic wave

This equation means that the cumulative flow is either determined by the upstream conditions or the downstream conditions depending in which traffic state use. In terms of notation, we separate the Newell's equation 7 into the uncongested $N_1(x,t)$ and the $N_2(x,t)$ congested part. So, we introduce

$$N_1(x,t) = F(t - \frac{x}{V}) + n_0, \tag{8}$$

and

$$N_2(x,t) = G(t - \frac{l-x}{W}) + (l-x)K. \tag{9}$$

2.2 Estimation of vehicle trajectories with FIFO violation

Here, we introduce a quick review of the method developed in (Rey and Jin, 2016). This method is based in the following two properties explained in the mentioned paper:

2.2.1 Properties

Firstly, the density is between 0 and k_C along $N_1(x,t)$ and between k_C and K along $N_2(x,t)$.

$$-K \leq \frac{\partial N_2(x,t)}{\partial x} \leq -k_C \leq \frac{\partial N_1(x,t)}{\partial x} \leq 0.$$

Secondly, the road segment used in the model $x \in [0,l]$ can be divided into three sub-segments as shown in the figure 5. In general, $[r(i), e(i))$ represents the interval when the vehicle runs into free flow velocity. Besides, $[e(i), d(i)]$ represents the part of the segment where the uncongested and congested region coexists or intersects for a vehicle i. The last part, $(d(i), e(i)]$ is the one in which the congested conditions are presented. The second region $[e(i), d(i)]$, can also be represented as a unique time point and not an interval as it is done in the example 5. This time is called $e(i)$ and it is defined as the time when free flow condition ends. Therefore, there is only two segments: the first represents where uncongested zone rules, instead the second is when congested zone rules.

1. For $x \in [r(i), e(i))$, $N_1(x,t) < N_2(x,t)$, and $n(x,t) > n(e(i),t)$;
2. For $x \in [e(i), d(i))b]$, $N_1(x,t) = N_2(x,t)$, and $n(x,t) \in [n(d(i),t), n(e(i),t)]$;
3. For $x \in (d(i), s(i)]$, $N_1(x,t) > N_2(x,t)$, and $n(x,t) < n(d(i),t)$.

Following this, when $e(i) = d(i) = 0$(all in congested conditions) or $e(i) = d(i) = l$(all in uncongested).

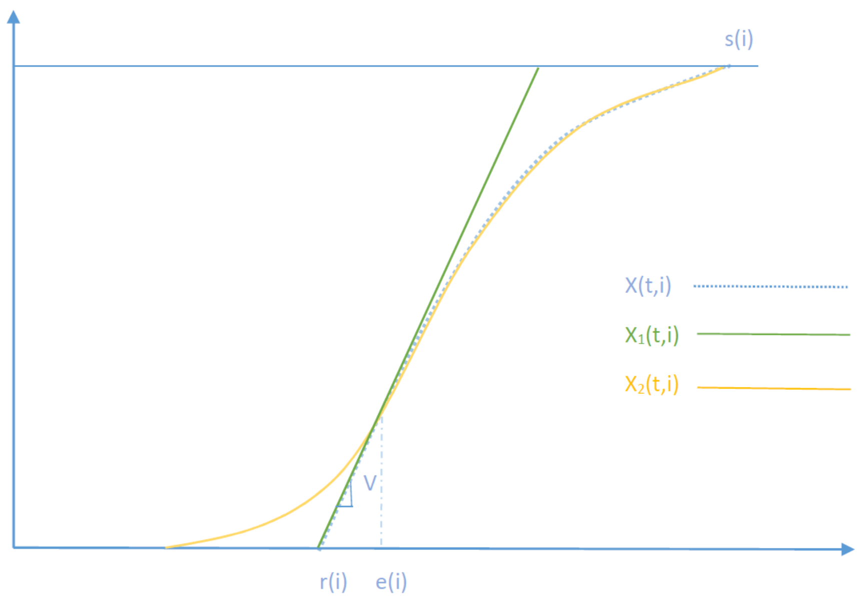


Figure 5: The three regions of the estimation trajectory

2.2.2 Trajectory estimation for the overtaking scenario

At this point we are able to make the first definitions. As we said we assume we have the information provided from the loop detector, as well as, the information of the entry and exit time of the vehicles provided by the vehicle reidentification.
The identification of the vehicle is done in an increasing order depending in their entrance time. Also we define, for a vehicle i, it's entrance time as r_i and exit time as s_i, that also corresponds to $X_i(r_i) = 0$ and $X_i(s_i) = l$. Besides, the vehicle i's order is defined by the $\theta_i(t)$ function.
So, taking into account all this, we can define the vehicle i's order at r_i is:

$$\theta_i(r_i) = i. \tag{10}$$

The order function has to be defined. In a hypothetical case where FIFO is respected during the whole segment, the i order will be constant. Instead, in the FIFO violation it has to be defined a better fit for the order function, that will be treated further.
Once we have the order function defined, we can denote the vehicle i's location by $X_i(t)$:

$$n(X_i(t), t) = \theta_i(t). \tag{11}$$

That means that the number of vehicles which have passed at the position where vehicle i is corresponds to his actual order represented by the function $\theta_i(t)$.So, if $X_i(t) < X_j(t)$ for vehicles i and j, then $\theta_i(t) > \theta_j(t)$ because j has entered before into the segment.

Combining the location definition 11 with 7 we are able to redefining the equations as:

$$N_1(X_i^1(t), t) = F(t - \frac{X_i^1(t)}{V}) + n_0 = \theta_i(t), \tag{12}$$

and $X_i^2(t)$ satisfies

$$N_2(X_i^2(t), t) = G(t - \frac{l - X_i^2(t)}{W}) + (l - X_i^2(t))K = \theta_i(t). \tag{13}$$

$X_i^1(t)$ and $X_i^2(t)$ represent the value of the location in uncongested and congested environment respectively. Analytically, $X_i^1(t)$ and $X_i^2(t)$ are denoted as the respective inverse functions of $N_1(x,t)$ and $N_2(x,t)$. Trough the first of the properties mentioned before, both $N_1(x,t)$ and $N_2(x,t)$ are strictly decreasing,then $X_i^1(t)$ and $X_i^2(t)$ are well-defined as a inverse function.

Finally, the most important equation provided by (Rey and Jin, 2016) is the following: At t, the location of the ith vehicle on the road segment, $X_i(t)$, is given by

$$X_i(t) = \min\left\{X_i^1(t), X_i^2(t)\right\}. \tag{14}$$

2.2.3 Lineal approximation for the order-change function

Due to the great results as an approximation in (Rey and Jin, 2016), the lineal order function is also considered here. Thanks to Loop detectors and the reidentification method we have information of the position of the entrance and also from the exit. M_i is defined as the order variation at the end of the segment. It is positive if the vehicle is slower as the average and negative if it is faster as the others(it is increasing his order as overtaking other vehicles). So, the order function at the exit time s_i is defined as:

$$\theta_i(s_i) = i + M_i. \tag{15}$$

Then, as we said, although it is a staircase function we suppose a linear order changing model defined as:

$$\theta_i(t) = a_i t + b_i. \tag{16}$$

With the boundary conditions of this equation: 10 at the entrance, and 15 at the exit point it is defined the order-changing function:

$$\theta_i(t) = a_i(t - r_i) + i, \tag{17}$$

where

$$a_i = \frac{M_i}{s_i - r_i}. \tag{18}$$

Now, we are able to define the equations that guide as to the trajectory estimation:

$$F(t - \frac{X_i^1(t)}{V}) + n_0 = \theta_i(t), \tag{19}$$

$$G(t - \frac{l - X_i^2(t)}{W}) + (l - X_i^2(t))K = \theta_i(t). \tag{20}$$

Here it is important to take into account the propriety 14, because although each of both equations will return a position value, the minimum has to be taken.The other values needed are the cumulative flows $F(t), G(t)$, the order-changing function 17, the inital number of vehicles n_o and the estimated values from the fundamental diagram(V, W, K).

3 Traffic estimation with estimated trajectory and overtaking effects

The goal of this study is to estimate the traffic state by using the cumulative flow within a road segment where FIFO is violated. Count flow information of both entrance and end is given, as well as, the estimated trajectory traveled through the segment. So, the input data is:

- cumulative flow functions: $F(t)$ and $G(t)$ for t>0,
- $X(t,i)$: trajectory functions of I vehicles, where $i = 1, 2, ...I$
- Overtaking lineal effects consideration:$\theta_i(s_i) = i + M_i$

For a vehicle i, we can define the following equations which will you to find the parameters that minimizes them:

The first equation is based in the information that we have from the data, it is like a conservation equation in terms of order.

$$F(r(i)) + n_o = G(s(i)) + M_i \tag{21}$$

where, M has a different sign that the one that had before.

Then, the other two correspond to the approximation equation from Newell 7.

$$N_1(e(i), X(e(i), i)) = F((e(i) - \frac{X(e(i), i)}{V}) + n_o = \theta_i(e_i) \tag{22}$$

$$N_2(e(i), X(e(i), i)) = G(e(i) - \frac{l - X(e(i), i}{W} + K(l - X(e(i), i) = \theta_i(e_i) \tag{23}$$

As we will see later, in this case the time in which we study this equation is $e(i)$, defined in 2.2 .This scenario is selected because it will help in the resolution.
So, the parameters we want to estimate are the followings: $\hat{n_o}, \hat{V}, \hat{W}, \hat{K}, \hat{e}(1), ..., \hat{e}(I)$. To do that we define Z as a function of all the parameters. Z has to be optimized looking for the

minimum error through a least square method.It is noteworthy that we face a non-linear case.

$$\min Z\big(\hat{n_o}, \hat{V}, \hat{W}, \hat{K}, \hat{e}(i), ..., \hat{e}(I)\big) = \sum_i^I \xi^2 + \sum_i^I \epsilon^2 + \sum_i^I \gamma^2$$
$$= [\hat{n_o} + F(r_i) - M_i - G(s_i)]^2 + [F(e(i)) - \frac{X(e(i), i)}{V} + n_o - \theta_i(e_i)]^2$$
$$+[(G(e(i) - \frac{l - X(e(i), i)}{\hat{W}}) + \hat{K}(l - X(e(i), i)) - \theta_i(e_i)]^2 \tag{24}$$

So,(W,K,V) are unknown variables in the fundamental diagram and n_o represents the initial condition of the road segment.

3.1 Solution using a optimization method

The objective function 25 can be faced using a decoupled method minimizing in parallel. Besides, we will define the free flow velocity V as the speed limit of the road, because is such an accurate approximation and we simplify the problem. Finally, will also consider the following simplification related with $e(i)$.

3.1.1 Simplified problem previously finding $e(i)$

As it has been defined, $e(i)$ is the last time instant in which the vehicle movement is still in free flow. So, in order to simplify the problem we can find first the values of $e(i)$.
There are three observed types of trajectories: 1) The vehicle moves the whole segment in free flow with a velocity equal or higher than the V defined. 2) At the beginning the movement is in free flow(as the first type) but at the end the vehicles reach the top of the segment with a lower velocity than V. 3) The whole segment is in congested condition, therefore the velocity is lower than V. So, taking into account this definition the only type in which $e(i)$ exists(have a value between r_i and s_i) is the type 2. This is shown through the blue point in the figure 6.

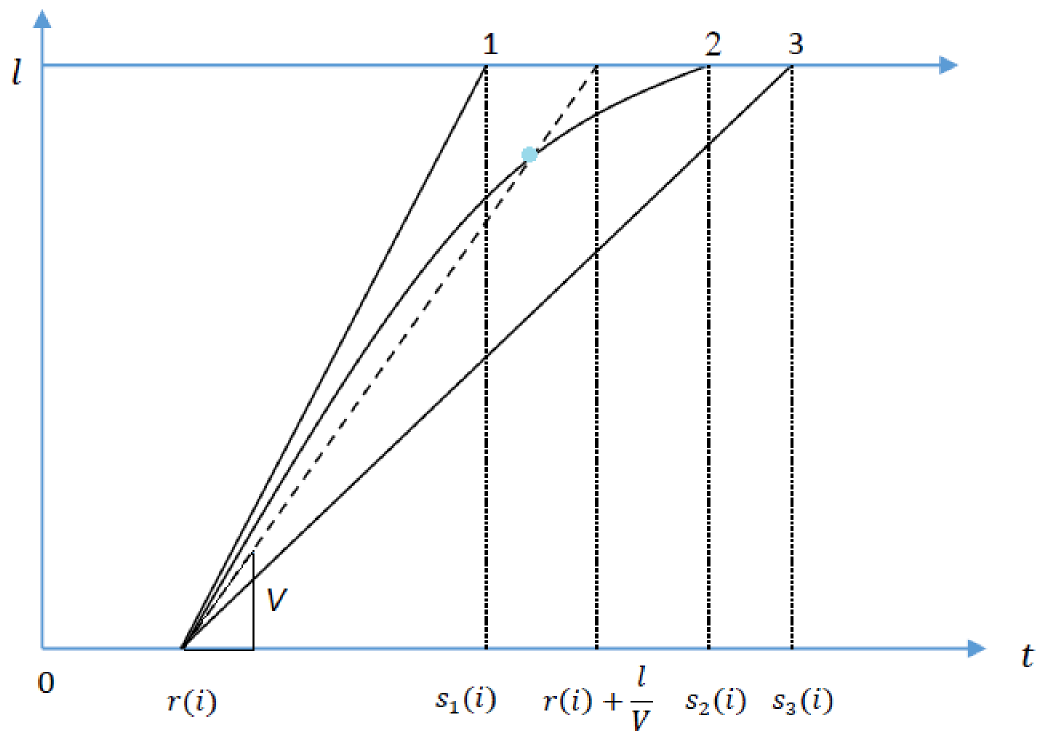


Figure 6: Typical three types of trajectories

3.1.2 Resolution

So, with the mentioned considerations the problem is simplified as:

$$\begin{aligned}\min Z\left(\hat{n_o}, \hat{W}, \hat{K}\right) &= \sum_i^I \xi^2 + \sum_i^I \gamma^2 \\ &= [\hat{n_o} + F(r_i) - M_i - G(s_i)]^2 + \\ [(G(e(i) - \frac{l - X(e(i), i)}{\hat{W}}) &+ \hat{K}(l - X(e(i), i)) - \theta_i(e_i)]^2\end{aligned} \tag{25}$$

We will face the problem using a decoupled method because the parameters are different in each sum.

Firstly, solving $\sum_i^I \xi^2$ is quite straightforward if we equal to zero the derivative respective to n_o of the sum.

Secondly, to solve the $\sum_i^I \gamma^2$ we use the Gauss-Newton method to solve the non-linear least square problem.
Starting from the initial guess $\vec{\theta}^{(0)} = (W^{(0)}, K^{(0)})^T$, the method updates results by iterating

$$\boldsymbol{\theta^{(j+1)}} = \boldsymbol{\theta^{(j)}} - [\boldsymbol{J(\theta^{(j)})^T J(\theta^{(j)})}]^{-1} \boldsymbol{J(\theta^{(j)})^T} \boldsymbol{\gamma}(\boldsymbol{\theta^{(j)}}),$$

where $\boldsymbol{J(\theta)}$ is the Jacobian matrix of $\boldsymbol{\gamma}(\boldsymbol{\theta})$, defined as

$$\boldsymbol{J(\theta)} = \begin{bmatrix} \frac{\partial \gamma_1(\boldsymbol{\theta})}{\partial W} & \frac{\partial \gamma_1(\boldsymbol{\theta})}{\partial K} \\ \vdots & \vdots \\ \frac{\partial \gamma_N(\boldsymbol{\theta})}{\partial W} & \frac{\partial \gamma_N(\boldsymbol{\theta})}{\partial K} \end{bmatrix}_{N \times 2}, \tag{26}$$

where the derivatives respective the parameters are,

$$\frac{\partial \gamma}{\partial \hat{W}} = G'(e(i) - \frac{l - X(e(i), i)}{\hat{W}})(\frac{l - X(e(i), i)}{\hat{W}^2}) \tag{27}$$

$$\frac{\partial \gamma}{\partial \hat{K}} = l - X(e(i), i) - G(s(i)) \tag{28}$$

The main idea is to approximate the Hessian matrix of the objective function, $\boldsymbol{\gamma^T(\theta^{(i)})\gamma(\theta^{(i)})}$, with its first order approximation $\boldsymbol{J(\theta^{(i)})^T J(\theta^{(i)})}$ at step i. The iteration stops after reaching enough precision. The tolerance of such method is about 10^-4.

4 Iterative method

Here, we introduce the iterative method developed to increase the accuracy of the trajectory estimation. This model is different from other previous studies because it's not necessary to have the values of the fundamental diagram to find a trajectory estimation. Applying these new approach we don't need these as a initial conditions because, as shown in the figure 7, with a initial guess of these values we can start to run the iteration until it reaches a convergence.

Besides, the cumulative flows and the overtaking function are need it in both methods as initial conditions, so it has to be the data we extract from the environment. However they are input data extracted from the traffic systems so don't change in the iteration process.

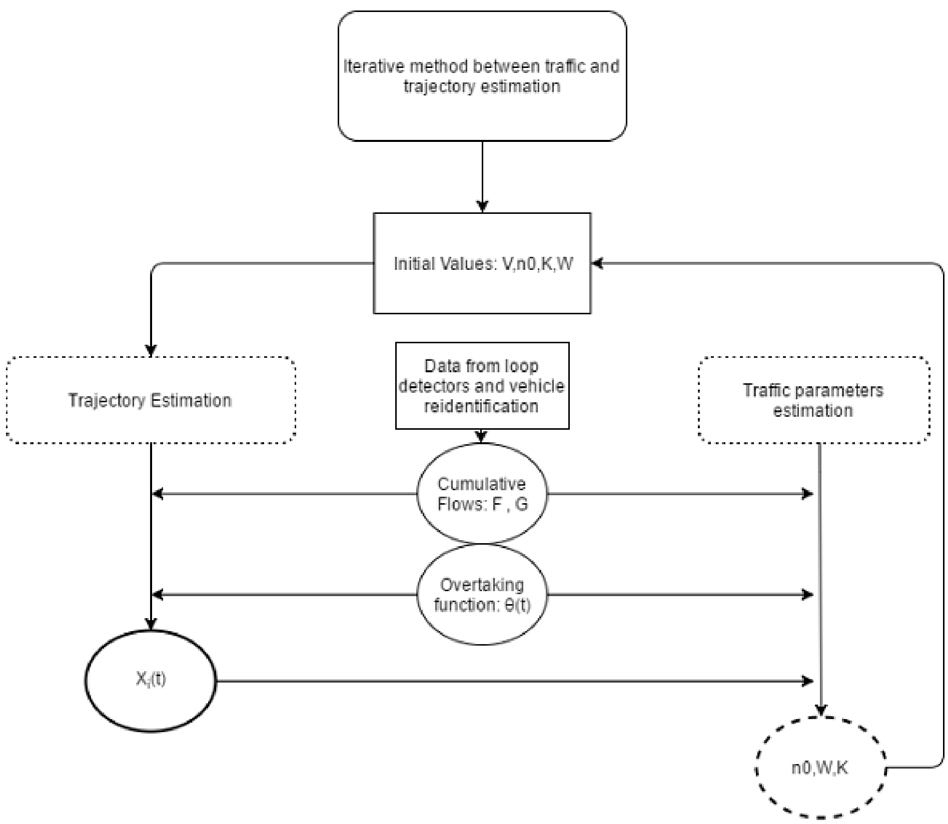


Figure 7: Flow chart from the iterative method

4.1 Convergence

Here we analyse the convergence properties that the model presents. We want to iterate the process until the values of W and K reaches a tolerance value.

As is shown in table 2, the method normally reaches a certain valour of convergence between four and five iterations in terms of error .However, our aim is to define a tolerance because the estimation process doesn't consider the real trajectory, and the error can't be considered as a convergence indicator. Even though,it is a good pointer of how many iterations are needed to get accurate results. Through the observation of the results we are able to indicate the following tolerances as the ones which the results are accurate enough.

So, the ratio tolerances are:

$$\frac{|\hat{W}_i - \hat{W}_{i+1}|}{\hat{W}_i} \leq 0.001 \tag{29}$$

$$\frac{|\hat{K}_i - \hat{K}_{i+1}|}{\hat{K}_i} \leq 0.005 \tag{30}$$

That way, when the ratio is under the tolerance value the iteration process ends.

5 Introduction of the autonomous vehicle in the trajectory estimation

The future of the vehicles are strictly related with the implementation of the autonomous vehicles(AV's). This kind of cars will be equipped with a whole set of technologies which will improve the performance of the vehicles in all senses.
In this study we assume that the autonomous vehicle has the following properties:

- Due to the GPS high accuracy localization we suppose known the trajectory of the AV for any time: $X(t, AV)\ \forall t$

- Due to the radars implemented in the car, it is able to know when the car overtakes or is being passed for another vehicle. That information leads us to a better fit order-change function of the AV's.

- Finally we assume AV's have the same vehicle identification technology that is set at the boundaries of the segment. This information will provide us which vehicles interact with the autonomous vehicles.

This three assumptions guides us to two consequences that will helps us in the better definition of the trajectory estimation.

5.1 Order function

Here, we develop a method to create a more accuracy order-change functions. With the information provided by the AV's we are able to know when(time frame) and which car is interacting with the AV. In the example provided by the figure 8, where some random trajectories are shown, we observe the definition of these times, t_k where k is number of interactions.

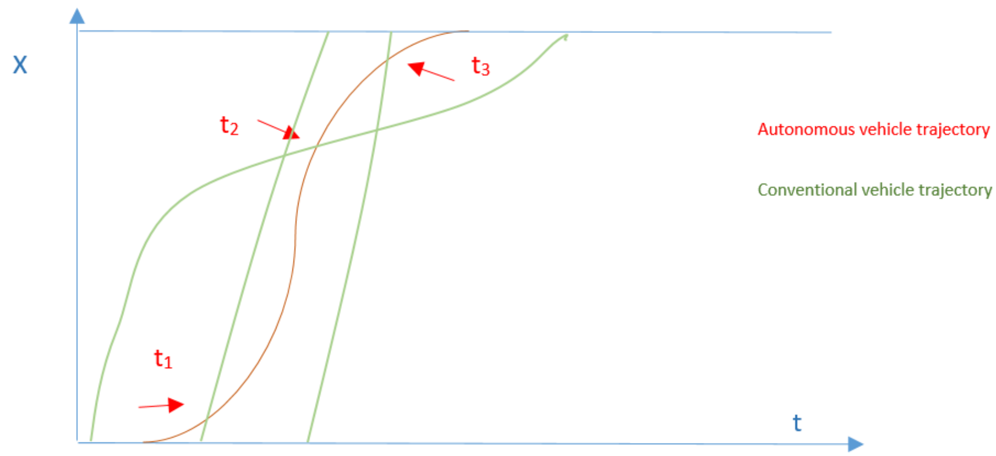


Figure 8: Mixed trajectories from both AV'S and CV's

Once we know t_k we can state two properties:

1. Supposing the two cars interacting are i and j and k the time they do, then: $X_i(t_k) = X_j(t_k)$

2. t_k gives straightforward information from the conventional car order-change function we wish to estimate, providing the value of $\theta_i(t_k)$. Therefore, we are able to create a better approximation.

We propose to estimate the function through a parabola from the three known points. That way we are able to have a continuous and derivable change-order function in all the time interval $[r_i, s_i]$. Taking into account i is a conventional car and j the AV, the three points known are:

$$\theta_i(r_i) = i \tag{31}$$

$$\theta_i(t_k) = \theta_j(t_k - \triangle T) = N_j \tag{32}$$

$$\theta_i(s_i) = i + M_i \tag{33}$$

Where $\triangle T$ is a time step, which wants to represent a time frame before the time when both vehicles intersect. Finally, applying these three points into the 2n order equation we obtain the following change-order function:

$$\theta_i(t) = at^2 + bt + c \tag{34}$$

where,

$$a = \frac{\frac{-M_i}{r_i^2 - s_i^2} - \frac{(i-N_j)(r_i-s_i)}{(r_i-t_k)(r_i^2-s_i^2)}}{1 - \frac{(r_i^2-t_k^2)(r_i-s_i)}{(r_i-t_k)(r_i^2-s_i^2)}}$$
$$b = \frac{i - N_j - a(r_i^2 - t_k^2)}{r_i - t_k}$$
$$c = i - ar_i^2 - br_i$$

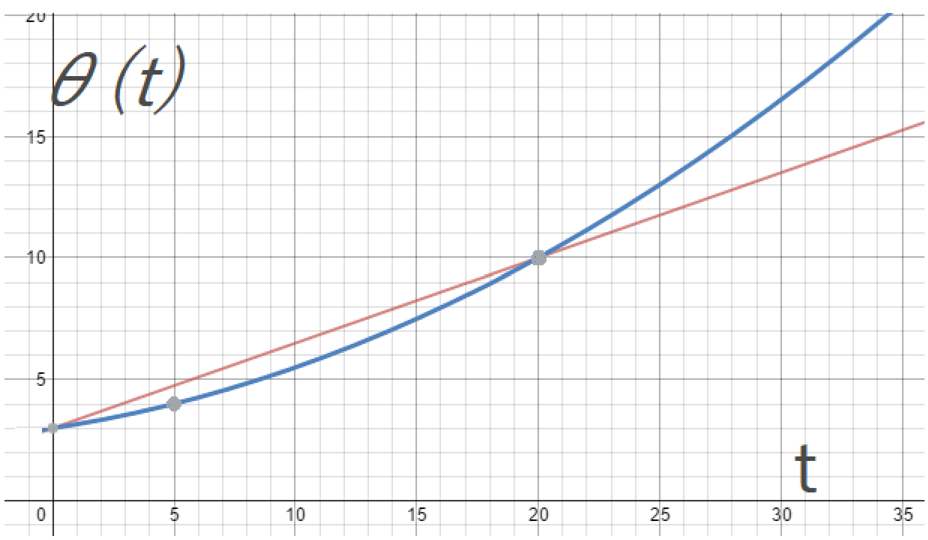


Figure 9: Comparison between lineal and parabolic order-change function

6 Implementation and Results

In order to verify the accuracy of the method proposed we implement it using data extracted from the *Next Generation Simulation*(NGSIM). The data is composed of the trajectories of vehicles travelling on the 101 freeway in Los Angeles, CA, from 7:50 AM to 8:35 AM on June 15, 2005.

The study segment is about 0.13 miles with five regular lines. Specifically, it is a stretch of the mentioned 101 freeway between the Ventura Blvd and the Cahuenga Blvd off-ramps. The data was recorded during a 45 minutes period, but it is split into three 15-minute intervals. Moreover, it has to be taken into account that the we start to count the vehicles after 2 minutes. This is done in order to make sure that all the vehicles in the segment have been tracked through the n_0 parameter.

6.1 Data treatment

We allow for that this data has been recorded without any loop detector or vehicle reidentification technology because it's not implemented in the road. As a result, we have to extract and simulate this information we want from the data. The information needed is: the cumulative flows provided by the loop detector and the entry/exit time of each vehicle given by the reidentification system.

6.1.1 Reidentification vehicles system

In order to reproduce what this kind of system will do, we need to extract from the data the entry/exit values of the vehicles(r_i and s_i).To do that we make a linear interpolation from the vehicle's trajectory, as indicates the following expression:

$$r_i = j\Delta t + \frac{x_0 - X_i(j\Delta t)}{X_i(j\Delta t + \Delta t) - X_i(j\Delta t)}\Delta t,$$

where j satisfies $X_i(j\Delta t) \leq x_0 \leq X_i(j\Delta t + \Delta t)$. Similarly, the exit time is

$$s_i = j'\Delta t + \frac{x_l - X_i(j'\Delta t)}{X_i(j'\Delta t + \Delta t) - X_i(j'\Delta t)}\Delta t,$$

where j' is chosen such that $X_i(j'\Delta t) \leq x_l \leq X_i(j'\Delta t + \Delta t)$.

In this case, x_0 and x_l are known values representing the entry and exit location respectively.

6.1.2 Cumulative flows

From the data is quite easily to count the number of vehicles passing trough the boundaries. However, it has to be taken into account that both $G(t)$ and $F(t)$ are step functions if this procedure is follow. So, from the original step functions a linear approximation is created.

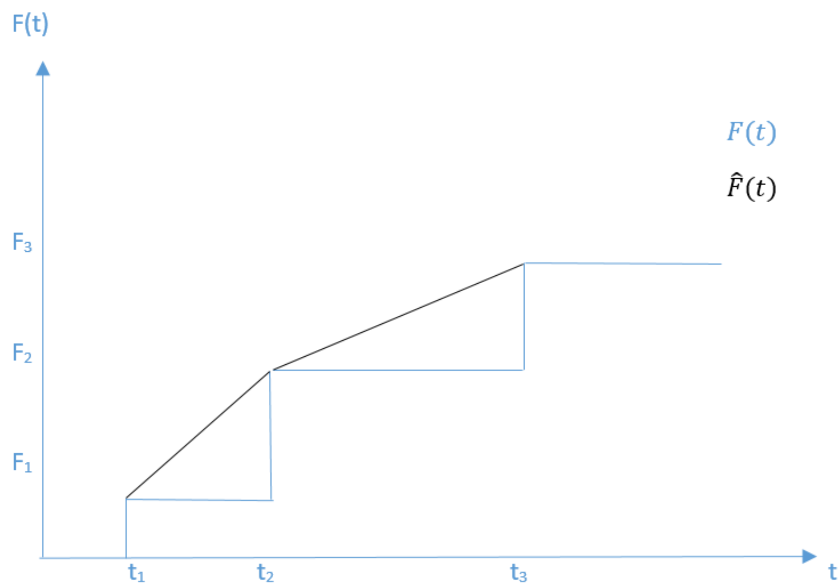


Figure 10: Linear approximation done with the cumulative flows

6.2 Results

Our aim here is to compare the trajectory estimated with the one provided by the data as is shown in the following example. In this case we observe the real trajectory and how the consequent iterations approaches it.

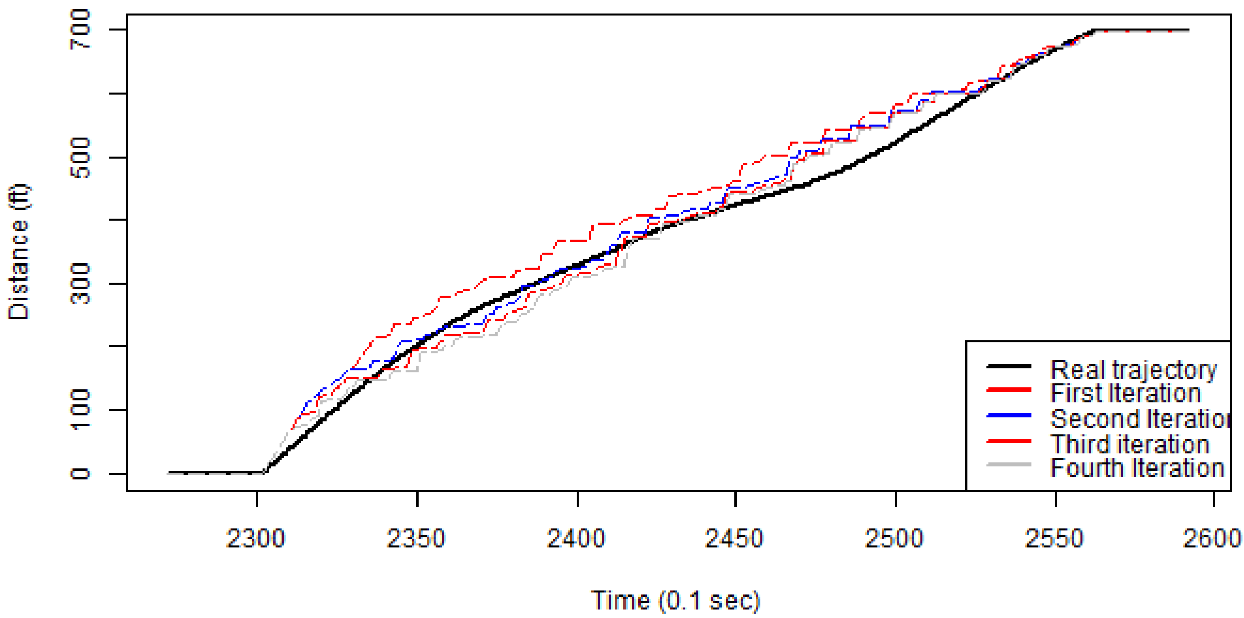


Figure 11: Trajectory estimation evolution during the iterative process

In order to examine which is the accuracy of the model proposed we compare both trajectories through the error between them. Defining $X_i(t)$ as the real trajectory provided by the NGSIM data and $\hat{X}_i(t)$ as the estimated trajectory, then the error is establish as the following difference:

$$Error_i = \frac{\int_{r_i}^{s_i} |\hat{X}_i(t) - X_i(t)| \, \mathrm{d}t}{\int_{r_i}^{s_i} |X_i(t)| \, \mathrm{d}t} \approx \frac{\sum_{j=r_i/\Delta t}^{s_i/\Delta t} |\hat{X}_i(j\Delta t) - X_i(j\Delta t)|}{\sum_{j=r_i/\Delta t}^{s_i/\Delta t} |X_i(j\Delta t)|}, \tag{35}$$

As we have seen in the figure 11 this error is calculated for each iteration in the i trajectory.

6.2.1 100 vehicles sample

We first make a selection of 100 vehicles that have entered into the segment and run the method with them. This small selection is done due to do a first test of the model while checking and analysing all the features without a huge amount of data. In the table 2 values obtained are displayed.

Iteration	n0[veh]	W[mph]	Tol.(W)	K[veh/mile]	Tol.(K)	Error
1	38.17	25	0	170	0	0.1293
2	39.26	19.73	0.21	181.38	0.067	0.0801
3	39.24	19.85	0.006	177.39	0.02	0.0634
4	39.20	19.74	0.005	175.99	0.0079	0.057
5	39.23	19.91	0.008	174.23	0.01	0.055
6	39.23	19.88	0.001	173.46	0.004	0.0542
7	39.21	19.90	0.001	172.79	0.004	0.0541
8	39.25	19.93	0.001	172.29	0.003	0.0541
9	39.23	19.90	0.001	171.98	0.002	0.0543
10	39.25	19.89	0.0005	171.66	0.002	0.0546

Table 2: Results from a set of 100 vehicles

Once we have the results we evaluated which is the performance of the tolerance ratio we have defined. As we can observe in the following figure 12, the tolerance is decreasing until values below the respectively thresholds defined in the section 4.1. Besides, we observe that the sixth iteration is the first one which is under the threshold, and from that value to the following there is a convergence event. On top of that fact, the error evolution is also stabilizing as we can see in the figure 13.

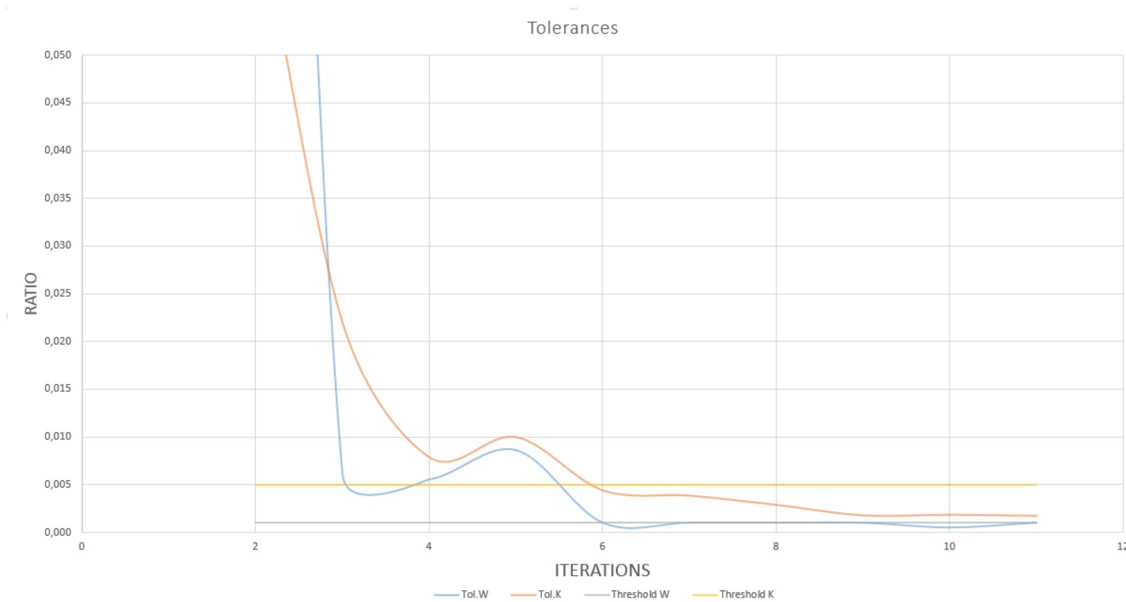


Figure 12: Tolerances evolution through the iterative process

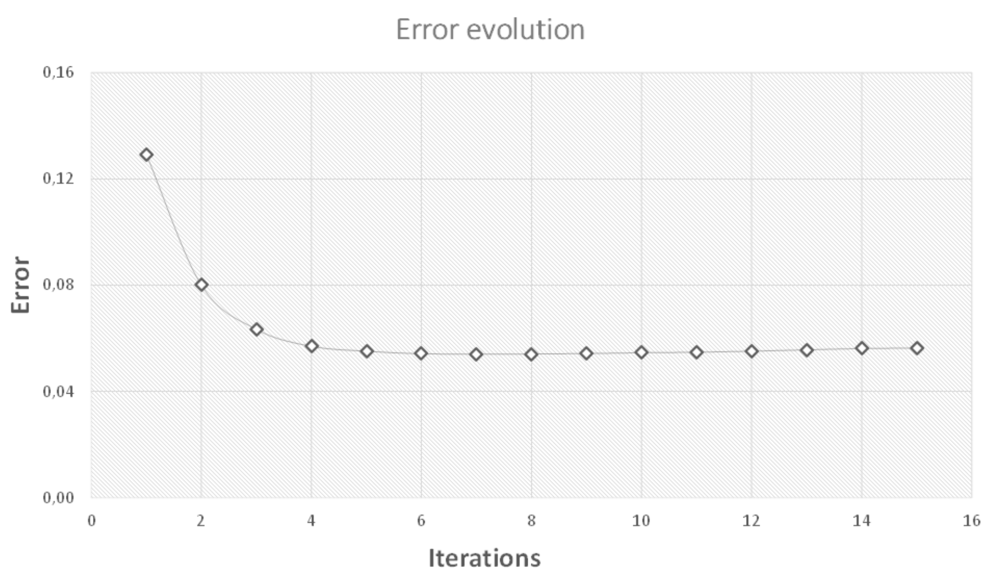


Figure 13: Error evolution through the iteration process

Finally, in the following figure 14 we can observe how the trajectories are being modified in each iteration. In this figure there is an accumulation of similar trajectories who represent the stabilization of the final trajectory. Despite it may not reaches a exact trajectory, the tolerance allows us to define a final trajectory with quite accuracy.

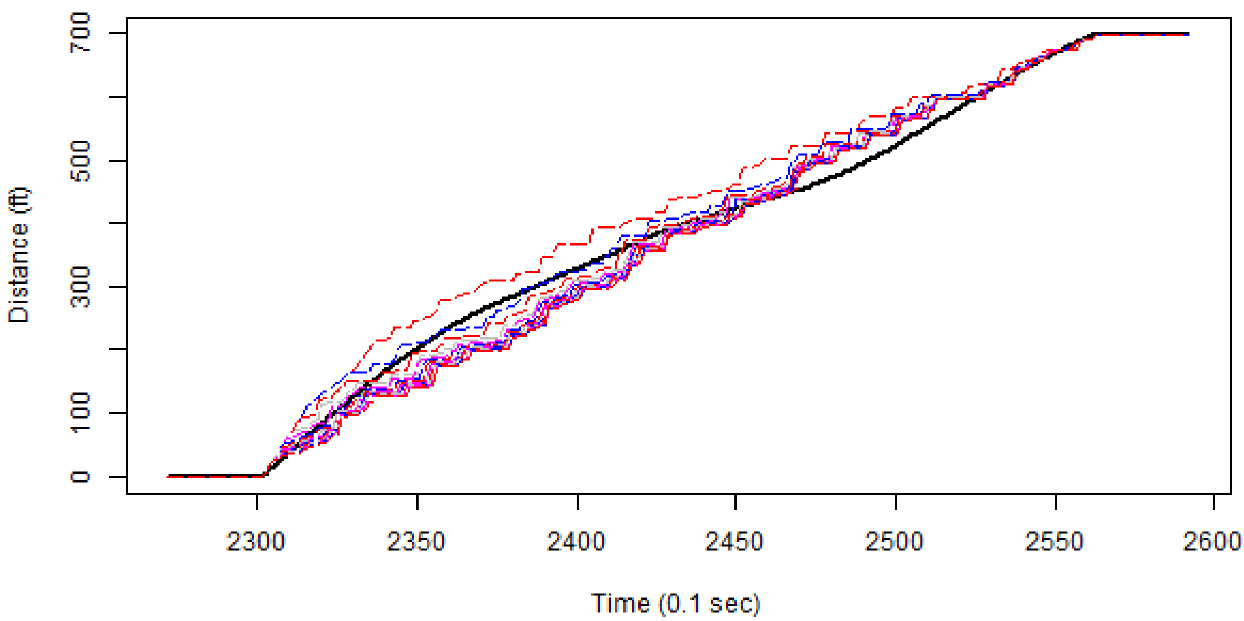


Figure 14: Trajectory estimation evolution during a whole iterative process

6.2.2 All data set implementation

Once we have seen that the model provides meaningful results we extend the sample. In this case we take all the vehicles of the first data set, in which there are up to 1800 vehicles. Trying to find a convergence result for all the vehicles may not give us an ambitious result because the parameters tend to an average. Instead, here propose a new solution in which we discretize the vehicles sample into 100 vehicles step. That way, we are trying to find better parameters estimation which can fit the actual conditions in a more realistic way.

A. Data Set 1

Set	Iterations	Error	no[veh]	W[mph]	K[veh/mile]
1	15	0.066	38.18	20.15	150.56
2	18	0.154	38.27	19.39	147.91
3	7	0.054	39.23	19.88	172.86
4	7	0.138	39.36	20.05	167.17
5	10	0.098	36.42	19.86	155.38
6	8	0.051	38.58	19.87	155.18
7	11	0.073	38.42	19.91	158.80
8	4	0.078	38.17	19.43	162.16
9	4	0.094	40.22	19.99	165.12
10	5	0.079	36.32	19.47	164.66
11	5	0.058	38.72	19.5	159.64
12	5	0.061	38.32	19.73	165.2
13	5	0.079	37.75	19.84	169.9
14	14	0.080	38.34	19.66	146.19
15	4	0.096	39.13	20.03	157.24
16	9	0.104	37.69	19.72	160.46
17	5	0.065	39.26	19.77	161.35
18	5	0.185	37.4	20.13	159.87
Average		0.088	38.32	21.5	157.8

Table 3: Results from the first whole set of samples

The final average error is about 0.88 which is about a 16 percent more accurate than the one provided without doing the iterative method.

B. Data Set 2

Set	Iterations	Error	no[veh]	W[mph]	K[veh/mile]
1	10	0,058	38,82	19,82	163,58
2	10	0,062	40,69	19,71	160,99
3	7	0,074	40,44	19,67	163,83
4	6	0,089	42,26	19,93	159,76
5	10	0,123	34,82	19,57	158,40
6	10	0,141	37,02	19,53	158,49
7	6	0,071	38,5	19,38	156,97
8	10	0,085	38,94	19,61	154,81
9	8	0,082	38,67	19,65	148,54
10	9	0,143	39,04	19,91	153,44
11	10	0,123	34,82	19,57	158,40
12	7	0,102	38,42	19,45	165,50
13	7	0,073	39,44	19,76	159,01
14	6	0,110	40,50	19,46	159,00
15	7	0,112	37,09	19,82	169,79
16	7	0,066	38,56	19,76	157,78
17	5	0,067	38,83	19,89	170,34
18	7	0,064	40,35	19,93	166,81
Average		0,091	38,73	19,69	160,30

Table 4: Results from the whole second set of samples

C. Data set 3

Set	Iterations	Error	no[veh]	W[mph]	K[veh/mile]
1	10	0,126	56,78	20,02	157,25
2	5	0,092	45,83	19,61	154,96
3	9	0,074	51,39	19,85	152,32
4	7	0,117	51,33	19,74	149,85
5	10	0,105	51,62	19,88	154,18
6	7	0,078	53,18	20,04	157,45
7	6	0,096	51,25	19,94	153,04
8	10	0,108	50,22	19,78	151,20
9	4	0,094	55,52	19,65	153,63
10	6	0,052	51,25	19,94	153,05
11	10	0,117	50,22	19,78	159,28
12	9	0,101	55,51	19,65	152,23
13	5	0,067	52,89	19,98	153,58
14	8	0,096	52,45	20,12	152,85
15	4	0,065	50,62	19,55	150,65
16	9	0,085	51,62	20,01	156,27
Average		0,092	51,98	19,85	153,86

Table 5: Results from the whole third set of samples

6.3 Results obtained with the AV's implementation

In this section we expose the results obtained with the method developed for the implementation of the AV's. In this particular case we have done the test in a 100 vehicle sample. To do the test, first we extract from the data the assumptions that we made in the section above 5. In the following figure 15 we can observe in black the AV's trajectory and in green and red the trajectories that interact with the driverless car.

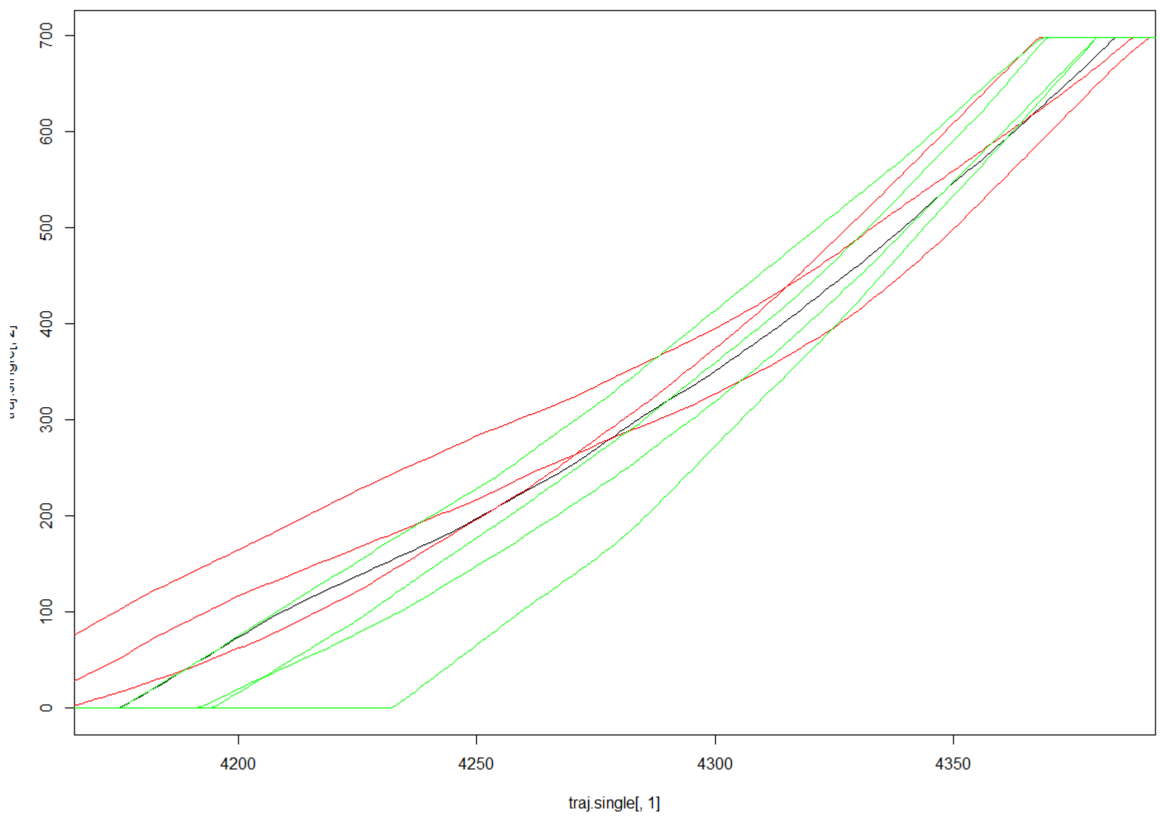


Figure 15: AV trajectory and their intersections

Then, we extract the information we need which is summarized in the following table 6. This information would be provided by the autonomous car in a real scenario.

Table 6: Add caption

Frame	i	Nj
4185	651	650
4188	649	651
4255	649	650
4277	646	651
4285	654	650
4350	652	651
4362	662	652
4366	644	653

Once we have this information we are able to find the new order-change function through the equation 34. This new order-function is parabolic and it is a better approximation to the real order-change step function. Finally, we need to retest all the method with the recent functions.
It is important to mention that the number of order function that will be modified is related with the number of vehicles interacting with the AV's. Besides, in this particular case we assume the relation AV/CV is 1/100, where CV is conventional car. With this particular relation the results are a little more accurate, but we expect a much better results with a higher autonomous vehicle MPR(market penetration rate).
In the following figure 16 we show of the trajectory is being modified for using a linear or a parabolic order-change function provided by the method developed. This is the particular case of a vehicle which overtakes the AV. In blue is represented the estimation with parabolic order-change function and in red the linear.

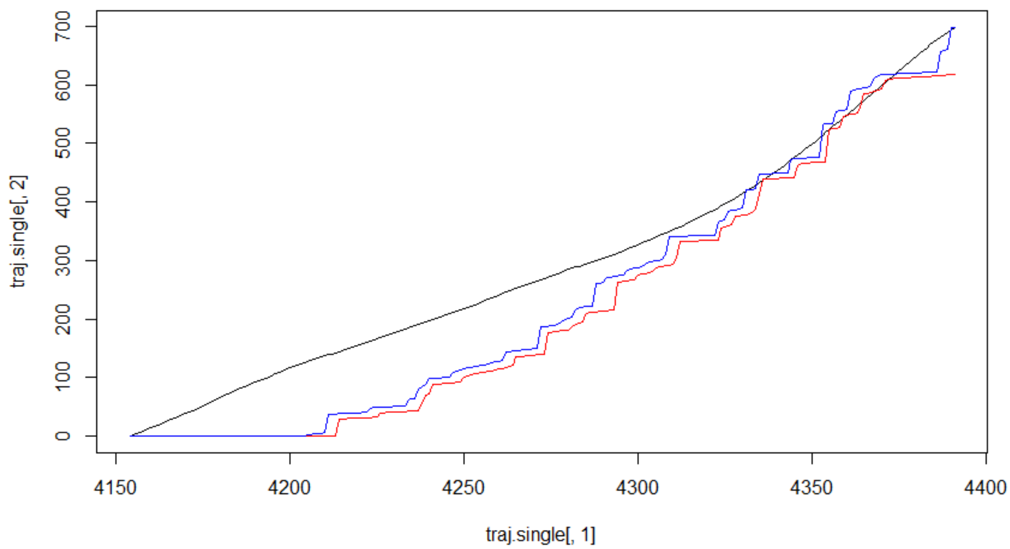


Figure 16: Trajectory estimation comparison between linear and parabolic order function

Finally we present the results obtained with and without the AV's implementation in the 100 vehicle set used. We present just 2 iterations to demonstrate that the use of the AV rapidly improve the performance of the estimation.

General Case				
1	0,275	38,2	25,0	170,0
2	0,156	38,5	19,8	165,6
With 1% AV's market penetration				
1	0,259	38,17	25	170
2	0,141	38,45	20	164,17

Table 7: Comparison between the results obtained in the two different procedures

7 Conclusions

The aim of this project was to fill the gap of adjusting the parameters (V,W,K) with the estimated trajectories in order to obtain a better estimated trajectory using an iterative method. The results in the following table 8 show that accuracy is slightly improved compared to other studies made using the same trajectory estimation method. However, there are still several error sources that lead us to the estimation errors: the cumulative flows $G(t)$ and $F(t)$ are step functions and they present some peak flows that make them grow faster than they should creating less smooth trajectories, as well as, there are wrong trajectories recorded in the NGSIM datasets.

Dataset	$E_1(\%)$	$E_2(\%)$	$Imp.(\%)$
1	10,52	8,8	16,35
2	9,53	9,1	4,51
3	9,88	9,2	6,88

Table 8: Summary of the comparison results

- E_1 represents the average error in the corresponding data set without using an iterative method.(Results provided by (Rey and Jin, 2016))
- E_2 represents the average error using the iterative method developed in this study
- $Imp.$ represents the error accuracy improvement in (%)

Moreover, using this method we achieve the goal of not needing the fundamental parameters of the road. The method allows us to find a solution with an initial guess and both Eulerian loop detector data and Lagrangian vehicle reidentification data. This fact results in a faster analysis of the road status, as well as, a method which can be applied in more scenarios. Besides, all the solutions are done in an overtaking environment, which represent a new methodology in the traffic flow theory as it is not following the FIFO principle.
Finally, with the AV's introduction we are able to analyse how in the near future the driverless cars will impact in the traffic systems. In this case, exploiting their powerful technologies allows us to developed a more accurate approximation by modifying the order-change function.

The results obtained are optimistic and lead us to think that more studies can be done in this direction. in order to take advantage of these cars and getting the most benefit of them in terms of traffic management.

Besides, we propose some further research in this study:

- Looking for a best definition of the $\theta(t)$ function for all the cases.
- Exploring the chance to include new technologies and their performance.
- Thinking in higher MPR rates for the AV's and how this would affect.
- Analysing the results using other parameters estimation

References

Barrios, C., Motai, Y., 2011. Improving estimation of vehicle's trajectory using the latest global positioning system with kalman filtering. IEEE Transactions on Instrumentation and Measurement 60 (12), 3747--3755.

Coifman, B., 2002. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. Transportation Research Part A 36 (4), 351--364.

Daganzo, C. F., 2005. A variational formulation of kinematic waves: basic theory and complex boundary conditions. Transportation Research Part B 39 (2), 187--196.

Daganzo, C. F., 1997. Fundamentals of transportation and traffic operations. SIAM, 26--27.

Greenshields, B. D., 1935. A study of traffic capacity. Highway Research Board Proceedings 14, 448--477.

Jin, W.-L., Li, L., 2007. First-In-First-Out is violated in real traffic. In: Proceedings of Transportation Research Board Annual Meeting.

Jin, W.-L., Zhang, Y., Chu, L., 2006. Measuring first-in-first-out violation among vehicles. In: Proceedings of TRB Annual Meeting.

Lighthill, M. J., Whitham, G. B., 1955. On kinematic waves. ii. a theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 229 (1178), 317--345.

Newell, G. F., 1993a. A simplified theory of kinematic waves in highway traffic, part i: General theory. Transportation Research Part B 27 (4), 281--287.

Rey, A., Jin, W.-L., 2016. Vehicle Trajectory estimation based on Newell's simplified kinematic wave model with eulerian and lagrangian traffic data

Sun, Z., 2016. Freeway Traffic Parameter and State Estimation with Eulerian and Lagrangian Data

USDOT, 2008. Ngsim: Next generation simulation.
URL http://www.ngsim.fhwa.dot.gov

Appendices

A The 201-300 vehicles interval from the data set 1

A.1 Estimated trajectories

Here we present the data obtained for a 100 vehicles set of the 18 we have done for the first 15 minutes data set.Specifically is the one defined between the vehicles 201 and 300.

ID	Entry Time	Exit time	Overtakes	e(i)	X(e(i))	Theta(e(i))
201	1963,497	2117,795	3	1965,278	11,731	239,187
202	1969,299	2116,114	1	1980,551	93,47	240,23
203	1969,454	2110,386	-3	1976,541	57,793	241,018
204	1977,008	2133,913	4	1991,635	119,884	242,514
205	1977,929	2138,859	5	1978,079	0,618	243,16
206	1978,517	2114,393	-5	1985,574	58,286	243,919
207	1983,96	2132,129	0	1984,073	0,356	245,158
208	1984,109	2125,824	-3	1995,258	90,403	245,927
209	1984,442	2128,537	-3	1988,586	31,529	247,083
210	1992,192	2151,783	1	1997,736	41,835	248,188
211	1993,996	2138,732	-2	1994,071	0,04	249,158
212	2002,315	2169,455	4	2008,463	47,669	250,294
213	2004,284	2153,578	-1	2014,531	83,927	251,092
214	2005,17	2170,487	3	2013,17	63,137	252,3
215	2005,841	2155,555	-2	2006,22	1,414	253,155
216	2014,528	2173,82	2	2016,475	12,765	254,176
217	2015,699	2164,918	-3	2018,401	20,722	255,111
218	2019,596	2177,013	1	2021,466	12,334	256,166
219	2019,841	2167,718	-4	2021,229	10,263	257,126
220	2022,825	2192,02	2	2023,083	1,402	258,16
221	2030,27	2192,71	2	2032,661	14,09	259,179
222	2036,322	2184,924	-2	2041,119	36,723	260,095
223	2038,769	2186,925	-2	2040,229	11,19	261,141

224	2039,194	2199,348	0	2042,578	22,409	262,158
225	2042,485	2214,567	3	2044,428	12,301	263,184
226	2046,934	2213,384	0	2047,082	0,581	264,158
227	2051,35	2214,238	0	2055,263	28,544	265,158
228	2054,496	2208,009	-3	2057,632	22,75	266,109
229	2060,167	2232,677	3	2063,268	19,342	267,207
230	2064,421	2228,304	1	2066,521	13,848	268,167
231	2064,969	2238,307	3	2065,085	0,268	269,158
232	2066,23	2223,559	-2	2068,076	0,767	270,135
233	2071,102	2235,868	0	2075,819	16,274	271,158
234	2073,952	2241,681	2	2077,054	9,799	272,194
235	2076,478	2223,411	-6	2077,215	2,471	273,136
236	2077,189	2251,288	2	2084,082	0,085	274,236
237	2085,529	2247,4	0	2088,078	1,278	275,158
238	2093,433	2240,461	-3	2098,091	36,553	276,064
239	2093,953	2261,679	2	2099,161	27,521	277,218
240	2095,752	2256,205	-1	2102,683	23,772	278,119
241	2097,617	2279,898	4	2102,486	13,377	279,254
242	2101,94	2274,306	1	2102,084	0,518	280,158
243	2106,358	2264,88	-1	2107,386	3,664	281,154
244	2109,052	2259,511	-4	2111,073	1,108	282,106
245	2115,504	2298,594	6	2116,447	3,493	283,174
246	2116,618	2301,886	6	2124,549	22,579	284,397
247	2118,784	2276,214	-3	2126,167	19,682	285,02
248	2121,665	2289,026	-1	2126,208	10,651	286,132
249	2130,005	2322,879	9	2135,219	37,405	287,391
250	2130,16	2287,212	-4	2133,904	16,785	288,085
251	2130,588	2292,414	-2	2139,143	28,032	289,054
252	2130,636	2289,308	-4	2137,672	5,368	289,997
253	2135,74	2297,109	-3	2142,763	15,081	291,041
254	2143,597	2314,316	0	2144,417	3,545	292,158
255	2145,676	2302,411	-2	2146,23	1,96	293,153

256	2145,895	2316,561	0	2153,08	0,426	294,158
257	2148,398	2354,599	7	2157,097	0,596	295,45
258	2152,395	2316,31	-3	2160,077	1,108	296,018
259	2156,316	2331,238	0	2166,081	1,278	297,158
260	2158,567	2317,083	-3	2171,11	19,001	297,922
261	2162,644	2371,49	8	2172,684	6,22	299,516
262	2169,809	2334,308	-2	2173,237	2,301	300,119
263	2170,505	2346,905	-1	2178,083	0,426	301,115
264	2180,076	2392,308	11	2181,362	3,323	302,205
265	2180,091	2356,015	0	2184,084	0,426	303,158
266	2184,734	2338,904	-5	2200,313	108,807	303,662
267	2188,929	2357,607	0	2190,246	2,471	305,158
268	2193,905	2416,405	14	2211,602	102,331	307,233
269	2196,751	2347,099	-6	2218,325	143,23	306,31
270	2198,68	2367,576	-2	2218,814	130,279	307,929
271	2201,688	2356,613	-5	2219,324	125,848	308,599
272	2206,151	2375,71	-2	2221,207	116,475	309,982
273	2210,487	2432,471	13	2211,488	4,154	311,188
274	2211,85	2381,745	-3	2212,083	1,351	312,155
275	2219,663	2390,644	-2	2231,606	101,99	313,025
276	2223,023	2452,702	15	2232,666	75,943	314,744
277	2223,615	2384,511	-5	2234,298	90,573	314,835
278	2227,387	2402,572	0	2236,63	75,792	316,158
279	2233,184	2404,282	0	2243,268	82,223	317,158
280	2234,56	2398,099	-3	2242,607	67,126	318,021
281	2234,576	2391,647	-7	2236,457	12,537	319,094
282	2237,878	2415,234	-1	2238,087	1,097	320,157
283	2239,051	2475,23	14	2240,519	4,733	321,214
284	2246,36	2413,008	-4	2255,888	78,133	321,95
285	2251,801	2498,695	17	2252,181	1,747	323,171
286	2251,827	2418,284	-3	2252,245	1,526	324,154
287	2256,053	2432,915	0	2269,461	110	325,158

288	2263,199	2429,809	-4	2279,319	134,88	325,778
289	2263,336	2514,822	17	2264,674	5,624	327,202
290	2263,887	2432,143	-5	2264,083	1,013	328,154
291	2270,344	2449,348	-2	2283,673	111,533	329,016
292	2272,031	2445,613	-4	2284,742	103,354	329,882
293	2276,568	2537,286	19	2284,2	62,915	331,699
294	2277,09	2392,634	-18	2291,42	118,35	329,991
295	2280,661	2449,827	-5	2290,49	83,416	332,882
296	2286,806	2465,039	-3	2296,702	85,461	334,003
297	2287,836	2466,997	-3	2288,087	1,472	335,155
298	2297,394	2478,243	0	2300,563	22,932	336,158
299	2298,065	2455,108	-7	2309,196	89,891	336,67
300	2301,486	2562,537	18	2305,122	27,57	338,4

A.2 Real trajectory in data set 1

Trying to show all the real trajectories would spend too much space and it wouldn't be worth it. That's way we only show the real trajectory form the 7 first vehicles(201-207) as the NGSIM data provided.

$\triangle T$	**Veh.201**	**Veh.202**	**Veh.203**	**Veh.204**	**Veh.205**	**Veh.206**	**Veh.207**
1963	0	0	0	0	0	0	0
1964	2,327	0	0	0	0	0	0
1965	6,948	0	0	0	0	0	0
1966	11,525	0	0	0	0	0	0
1967	16,047	0	0	0	0	0	0
1968	20,538	0	0	0	0	0	0
1969	25,028	0	0	0	0	0	0
1970	29,527	3,156	2,387	0	0	0	0
1971	34,027	7,682	6,736	0	0	0	0
1972	38,527	12,214	11,029	0	0	0	0
1973	43,031	16,729	15,271	0	0	0	0
1974	47,526	21,226	19,495	0	0	0	0

1975	52	25,722	23,752	0	0	0	0
1976	56,467	30,221	28,061	0	0	0	0
1977	60,942	34,721	32,393	0	0	0	0
1978	65,438	39,22	36,667	4,524	0,288	0	0
1979	69,942	43,731	40,829	9,044	4,427	2,309	0
1980	74,442	48,219	44,953	13,537	8,543	7,148	0
1981	78,942	52,659	49,098	18,029	12,593	11,997	0
1982	83,442	57,08	53,282	22,528	16,602	16,775	0
1983	87,941	61,52	57,498	27,028	20,63	21,412	0
1984	92,441	66,009	61,717	31,527	24,693	25,951	0,183
1985	96,942	70,521	65,961	36,026	28,758	30,5	4,708
1986	101,442	75,021	70,217	40,526	32,787	35,168	9,219
1987	105,942	79,521	74,458	45,03	36,783	39,978	13,716
1988	110,442	84,021	78,696	49,525	40,774	44,9	18,212
1989	114,942	88,521	82,944	54	44,804	49,771	22,711
1990	119,442	93,021	87,236	58,468	48,774	54,515	27,211
1991	123,941	97,52	91,611	62,943	52,612	59,334	31,711
1992	128,441	102,02	96,068	67,438	56,401	64,326	36,21
1993	132,941	106,52	100,57	71,942	60,239	69,447	40,71
1994	137,441	111,02	105,084	76,442	64,208	74,748	45,22
1995	141,941	115,52	109,588	80,942	68,239	80,002	49,709
1996	146,44	120,02	114,087	85,442	72,24	85,257	54,152
1997	150,942	124,52	118,586	89,941	76,239	90,527	58,577
1998	155,437	129,02	123,086	94,441	80,244	95,774	63,02
1999	159,924	133,52	127,586	98,941	84,248	100,905	67,509
2000	164,421	138,02	132,086	103,442	88,234	105,9	72,021
2001	168,964	142,52	136,586	107,942	92,174	110,772	76,521
2002	173,591	147,021	141,085	112,442	96,059	115,506	81,02
2003	178,294	151,517	145,585	116,941	99,922	120,159	85,52
2004	183,032	156	150,085	121,441	103,807	124,754	90,02
2005	187,793	160,492	154,587	125,941	107,747	129,348	94,52
2006	192,589	165,047	159,083	130,441	111,733	134,025	99,02

2007	197,455	169,739	163,569	134,941	115,738	138,84	103,519
2008	202,409	174,548	168,067	139,441	119,743	143,802	108,019
2009	207,446	179,493	172,607	143,941	123,742	148,816	112,519
2010	212,573	184,499	177,224	148,44	127,742	153,778	117,019
2011	217,789	189,516	181,932	152,94	131,742	158,593	121,519
2012	223,072	194,52	186,736	157,44	135,741	163,269	126,019
2013	228,384	199,515	191,622	161,942	139,741	167,869	130,519
2014	233,681	204,515	196,534	166,437	143,741	172,475	135,019
2015	238,959	209,546	201,42	170,923	147,741	177,123	139,519
2016	244,235	214,642	206,277	175,423	151,741	181,791	144,019
2017	249,498	219,818	211,152	179,973	155,741	186,47	148,521
2018	254,672	225,054	216,061	184,591	159,741	191,169	153,016
2019	259,684	230,314	220,972	189,252	163,741	195,902	157,499
2020	264,554	235,568	225,846	193,913	167,741	200,68	161,992
2021	269,281	240,795	230,682	198,588	171,741	205,505	166,547
2022	273,836	245,962	235,521	203,339	175,74	210,403	171,233
2023	278,391	251,042	240,389	208,235	179,741	215,361	176,06
2024	283,043	256,062	245,311	213,245	183,74	220,323	180,956
2025	287,809	261,084	250,294	218,4	187,74	225,245	185,836
2026	292,635	266,152	255,323	223,624	191,737	230,112	190,681
2027	297,483	271,246	260,334	228,859	195,74	234,952	195,552
2028	302,369	276,31	265,241	234,06	199,757	239,78	200,518
2029	307,322	281,325	270,023	239,213	203,779	244,572	205,582
2030	312,336	286,315	274,69	244,302	207,796	249,316	210,716
2031	317,355	291,306	279,268	249,307	211,799	254,039	215,882
2032	322,317	296,306	283,802	254,208	215,797	258,812	221,058
2033	327,173	301,307	288,302	258,996	219,796	263,693	226,232
2034	331,934	306,307	292,769	263,685	223,79	268,667	231,386
2035	336,664	311,307	297,219	268,285	227,782	273,683	236,502
2036	341,445	316,314	301,748	272,829	231,791	278,695	241,56
2037	346,332	321,307	306,458	277,375	235,849	283,692	246,572
2038	351,293	326,272	311,327	281,979	239,964	288,693	251,559

2039	356,242	331,225	316,347	286,681	244,105	293,711	256,541
2040	361,124	336,189	321,561	291,475	248,239	298,726	261,538
2041	365,959	341,183	326,785	296,359	252,363	303,682	266,571
2042	370,827	346,189	331,98	301,318	256,497	308,528	271,65
2043	375,787	351,189	337,175	306,321	260,641	313,246	276,744
2044	380,804	356,189	342,409	311,334	264,753	317,898	281,807
2045	385,796	361,189	347,661	316,341	268,795	322,471	286,823
2046	390,686	366,189	352,92	321,337	272,747	326,944	291,814
2047	395,45	371,189	358,223	326,324	276,62	331,389	296,804
2048	400,12	376,189	363,582	331,307	280,453	335,876	301,804
2049	404,742	381,189	368,984	336,294	284,303	340,449	306,805
2050	409,365	386,189	374,402	341,292	288,222	345,053	311,805
2051	414,039	391,189	379,82	346,294	292,202	349,651	316,812
2052	418,807	396,181	385,231	351,294	296,2	354,346	321,805
2053	423,692	401,159	390,642	356,293	300,209	359,19	326,771
2054	428,674	406,146	396,054	361,293	304,256	364,162	331,727
2055	433,702	411,189	401,466	366,292	308,375	369,283	336,693
2056	438,717	416,321	406,876	371,293	312,539	374,501	341,686
2057	443,678	421,503	412,289	376,294	316,705	379,737	346,693
2058	448,56	426,646	417,71	381,294	320,893	384,958	351,693
2059	453,353	431,703	423,133	386,293	325,081	390,171	356,694
2060	458,06	436,698	428,554	391,286	329,262	395,386	361,69
2061	462,677	441,681	433,968	396,272	333,509	400,601	366,672
2062	467,218	446,677	439,378	401,253	337,867	405,813	371,667
2063	471,714	451,679	444,789	406,23	342,35	411,032	376,729
2064	476,205	456,677	450,201	411,208	346,883	416,267	381,914
2065	480,717	461,682	455,612	416,201	351,388	421,509	387,208
2066	485,222	466,698	461,023	421,215	355,886	426,744	392,516
2067	489,664	471,702	466,435	426,22	360,386	431,963	397,748
2068	493,997	476,645	471,847	431,165	364,887	437,174	402,873
2069	498,226	481,475	477,259	435,995	369,383	442,389	407,954
2070	502,402	486,186	482,67	440,706	373,867	447,606	413,088

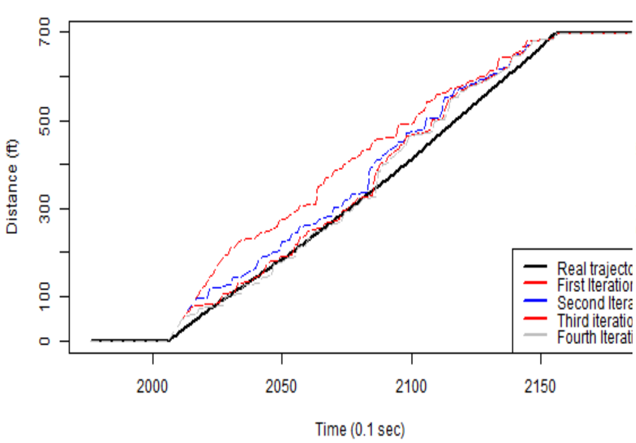
2071	506,553	490,859	488,081	445,379	378,364	452,815	418,34
2072	510,665	495,564	493,495	450,086	382,914	458,012	423,724
2073	514,716	500,306	498,909	454,83	387,561	463,223	429,185
2074	518,716	505,026	504,317	459,544	392,304	468,521	434,645
2075	522,702	509,656	509,709	464,149	397,084	473,968	440,039
2076	526,697	514,198	515,079	468,63	401,845	479,536	445,336
2077	530,699	518,695	520,432	473,024	406,577	485,254	450,542
2078	534,699	523,182	525,779	477,389	411,333	491,057	455,659
2079	538,699	527,677	531,127	481,77	416,175	496,876	460,699
2080	542,699	532,179	536,476	486,172	421,109	502,681	465,697
2081	546,7	536,679	541,83	490,546	426,101	508,481	470,684
2082	550,7	541,179	547,193	494,838	431,111	514,294	475,68
2083	554,7	545,679	552,546	499,048	436,117	520,105	480,682
2084	558,7	550,179	557,853	503,221	441,117	525,876	485,682
2085	562,7	554,678	563,069	507,396	446,12	531,558	490,68
2086	566,7	559,179	568,172	511,552	451,133	537,127	495,684
2087	570,7	563,679	573,187	515,587	456,136	542,608	500,7
2088	574,698	568,179	578,175	519,486	461,094	548,052	505,707
2089	578,699	572,675	583,189	523,258	465,969	553,491	510,653
2090	582,705	577,178	588,223	526,889	470,762	558,926	515,461
2091	586,714	581,696	593,251	530,475	475,508	564,335	520,152
2092	590,72	586,22	598,256	534,062	480,249	569,708	524,707
2093	594,721	590,738	603,253	537,66	485,006	575,094	529,2
2094	598,72	595,241	608,248	541,261	489,75	580,532	533,684
2095	602,72	599,738	613,232	544,861	494,458	585,998	538,179
2096	606,72	604,238	618,225	548,461	499,135	591,452	542,681
2097	610,72	608,738	623,279	552,06	503,803	596,864	547,181
2098	614,72	613,238	628,471	555,661	508,467	602,249	551,679
2099	618,72	617,736	633,781	559,261	513,087	607,627	556,184
2100	622,72	622,241	639,225	562,861	517,632	613,019	560,697
2101	626,72	626,755	644,722	566,461	522,118	618,499	565,2
2102	630,72	631,257	650,226	570,061	526,602	624,129	569,659

2103	634,72	635,703	655,748	573,662	531,15	629,882	574,038
2104	638,722	640,073	661,337	577,261	535,783	635,784	578,336
2105	642,716	644,402	667,024	580,861	540,452	641,775	582,556
2106	646,702	648,761	672,787	584,466	545,082	647,779	586,682
2107	650,703	653,208	678,59	588,073	549,636	653,767	590,739
2108	654,763	657,791	684,393	591,682	554,134	659,748	594,793
2109	658,927	662,497	690,156	595,299	558,62	665,732	598,907
2110	663,214	667,321	695,844	598,932	563,116	671,715	603,072
2111	667,605	672,293	698	602,58	567,617	677,698	607,204
2112	672,053	677,334	698	606,232	572,107	683,681	611,257
2113	676,516	682,387	698	609,884	576,617	689,664	615,256
2114	680,975	687,418	698	613,536	581,168	695,647	619,242
2115	685,44	692,429	698	617,178	585,735	698	623,237
2116	689,924	697,431	698	620,804	590,286	698	627,239
2117	694,422	698	698	624,458	594,796	698	631,241
2118	698	698	698	628,273	599,286	698	635,236
2119	698	698	698	632,238	603,786	698	639,219
2120	698	698	698	636,363	608,286	698	643,21
2121	698	698	698	640,639	612,786	698	647,279
2122	698	698	698	645,075	617,286	698	651,504
2123	698	698	698	649,538	621,786	698	655,858
2124	698	698	698	653,986	626,286	698	660,342
2125	698	698	698	658,424	630,786	698	664,979
2126	698	698	698	662,865	635,289	698	669,696
2127	698	698	698	667,305	639,78	698	674,443
2128	698	698	698	671,745	644,254	698	679,172
2129	698	698	698	676,185	648,756	698	683,841
2130	698	698	698	680,625	653,416	698	688,425
2131	698	698	698	685,065	658,224	698	692,934
2132	698	698	698	689,506	663,192	698	697,418
2133	698	698	698	693,946	668,311	698	698
2134	698	698	698	698	673,608	698	698

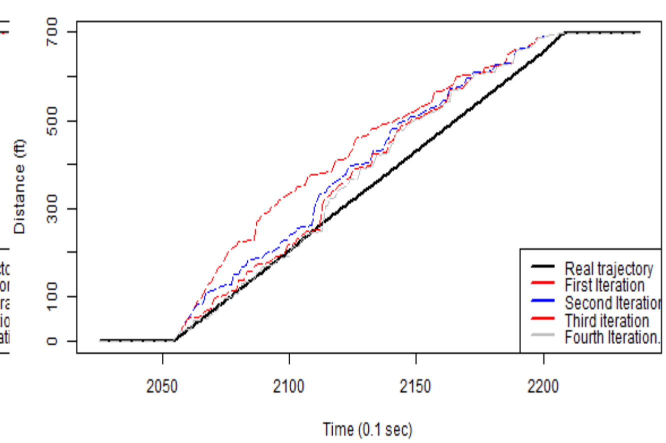
2135	698	698	698	698	678,904	698	698
2136	698	698	698	698	684,048	698	698
2137	698	698	698	698	689,059	698	698
2138	698	698	698	698	693,913	698	698
2139	698	698	698	698	698	698	698
2140	698	698	698	698	698	698	698
2141	698	698	698	698	698	698	698
2142	698	698	698	698	698	698	698

A.3 Graphics data set 1

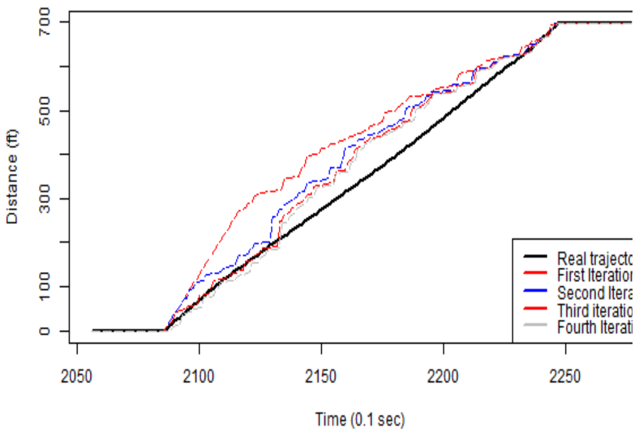
Finally we have selected randomly some of the graphics we obtained during the iterative method using the data presented above. In these graphics can be observed how the iterative method tends to approximate the estimation to the real trajectory. However, in this case the figures only show the first 4 iterations due to we don't pretend to overload the images with too much iterations and the process can be well seen with the first steps.



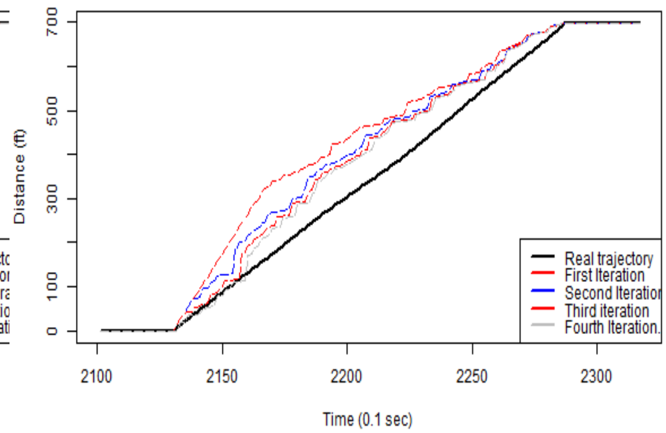
(a) Trajectory estimation veh.215



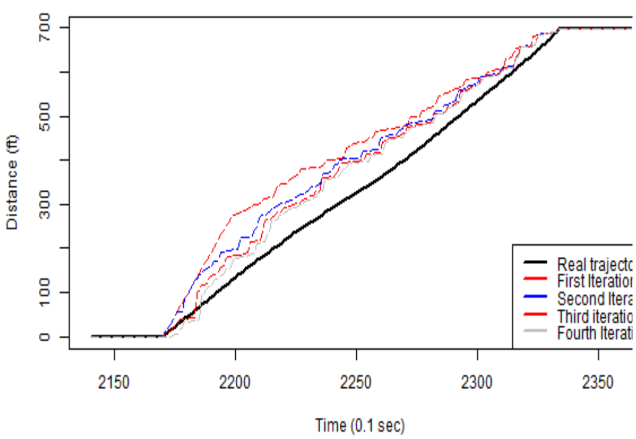
(b) Trajectory estimation veh.228



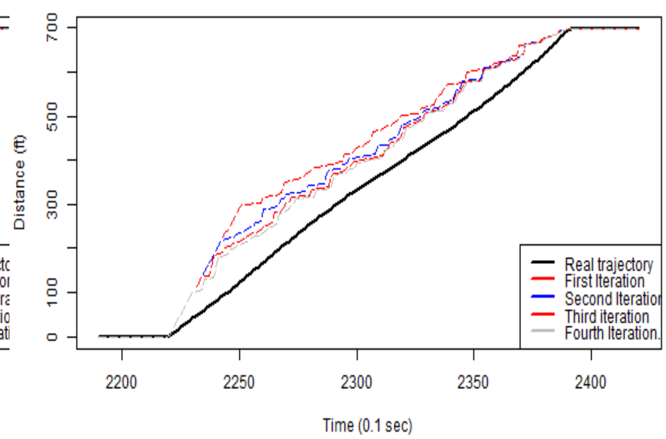
(a) Trajectory estimation veh.237



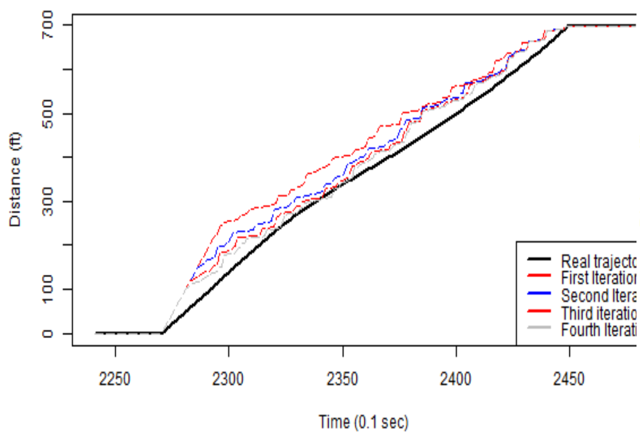
(b) Trajectory estimation veh.250



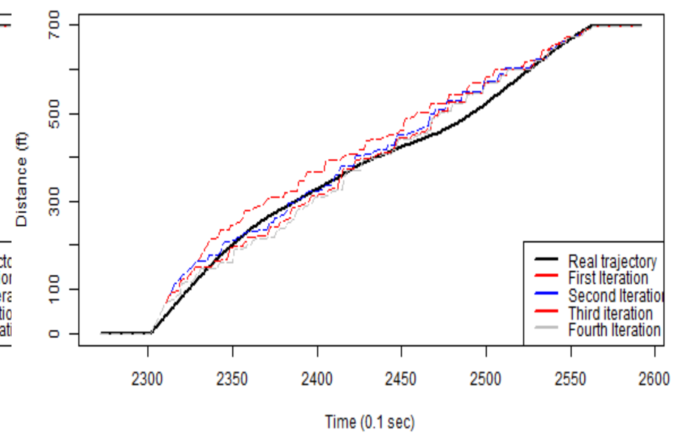
(a) Trajectory estimation veh.262



(b) Trajectory estimation veh.275



(a) Trajectory estimation veh.291



(b) Trajectory estimation veh.300

B The 601-700 interval from the data set 2

The same is done for a particular interval of 100 vehicles from the data set 2. Besides, some graphics showing the iterative method are also provided.

B.1 Estimated trajectory in data set 2

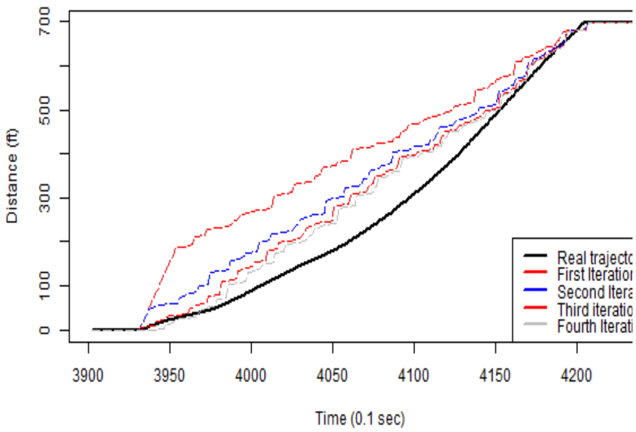
ID	Entry Time	Exit time	Overtakes	e(i)	X(e(i))	Theta(e(i))
601	3914,2	4175,78	-1	3916,06	0,09	639,5
602	3914,71	4241,29	14	3929,61	14,4	641,12
603	3920,1	4142,45	-10	3929,63	14,57	641,1
604	3929,02	4173,14	-5	3930,42	3,83	642,48
605	3931,43	4204,23	2	3937,07	0,09	643,54
606	3940,09	4261,36	15	3952,53	5,2	645,06
607	3945,28	4165,87	-9	3953,58	5,37	645,19
608	3960,28	4203,55	-2	3961,77	7,17	646,5
609	3964,38	4215,78	1	3965,19	2,13	647,51
610	3966,25	4280,54	15	3981,66	5,71	649,21
611	3969,2	4202,46	-6	3981,7	5,88	649,2

612	3972,08	4183,61	-9	3983,1	0,6	650,04
613	3989,34	4185,93	-9	3990,1	1,62	651,47
614	3990,79	4232,35	0	3998,06	0,6	652,5
615	3992,65	4295,55	16	4003,07	0,77	654,05
616	3994,44	4232,6	-1	4005,28	2,3	654,46
617	3998,85	4223,04	-6	4013,62	31,61	655,13
618	4002,64	4215,52	-9	4019,42	29,74	655,81
619	4013,72	4245,74	-1	4019,84	16,44	657,48
620	4013,85	4245,66	-3	4019,94	8,78	658,44
621	4020,58	4230,96	-8	4021,15	1,45	659,49
622	4024,17	4320,17	15	4039,15	18,49	661,26
623	4028,99	4261,16	-3	4041,23	20,02	661,35
624	4032,43	4287,12	3	4041,9	8,09	662,61
625	4035,29	4246,3	-6	4043,45	4,18	663,28
626	4035,37	4263,58	-4	4048,16	0,94	664,28
627	4053,27	4267,77	-4	4056,16	0,94	665,45
628	4054,96	4288,79	0	4059,17	1,45	666,5
629	4063,69	4279,95	-5	4070,65	23,77	667,36
630	4070,83	4330,77	10	4074,84	16,27	668,63
631	4078,64	4282,97	-5	4082,37	29,91	669,42
632	4079,65	4306,14	2	4080,39	3,01	670,51
633	4083	4315,72	3	4086,76	15,42	671,54
634	4087,22	4291,1	-5	4088,9	7,59	672,48
635	4100,05	4304,09	-2	4102,29	11,84	673,49
636	4109,26	4370,06	12	4113,52	30,95	674,68
637	4110,57	4335,74	4	4111,39	3,63	675,51
638	4112,42	4304,05	-6	4114,59	14,69	676,45
639	4119,34	4313,6	-4	4122,38	20,19	677,45
640	4122,04	4293,53	-10	4124,6	13,72	678,39
641	4123,83	4337,34	1	4124,16	1,44	679,5
642	4133,21	4323,4	-3	4134,47	3,83	680,49
643	4141,17	4320,3	-5	4146,31	37,41	681,37

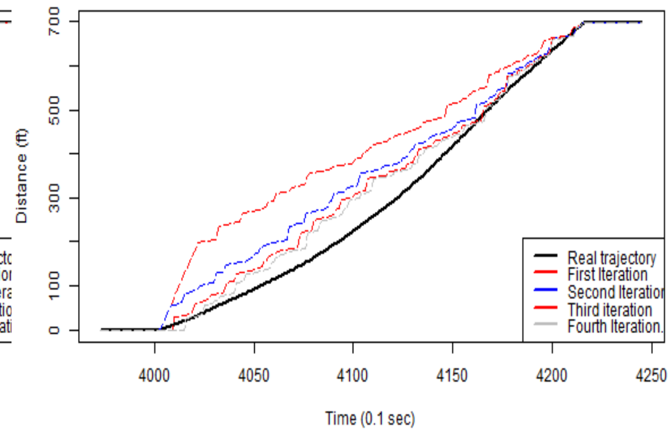
644	4142,21	4387,5	10	4149,14	27,86	682,78
645	4153,89	4373,04	4	4154,16	0,96	683,51
646	4154,38	4391	9	4155,64	5,21	684,53
647	4155,6	4355,58	-3	4159,47	30,44	685,45
648	4162,25	4341,96	-5	4166,48	30,93	686,4
649	4163,27	4367,77	-4	4167,69	22,92	687,43
650	4174,97	4383,76	2	4175,05	0,3	688,5
651	4175,3	4368,59	-5	4177,6	13,55	689,46
652	4191,63	4379,97	-2	4202,19	89,89	690,39
653	4192,67	4419,39	8	4193,28	2,66	691,52
654	4194,73	4369,46	-7	4197,48	21,66	692,41
655	4195,43	4403,32	3	4196,56	4,69	693,51
656	4199,68	4394,35	0	4203,35	29,19	694,5
657	4217,95	4441,8	9	4218,06	0,34	695,51
658	4222,36	4426,4	5	4229,56	57,91	696,67
659	4222,71	4403,97	0	4223,27	2,57	697,5
660	4223,33	4418,77	0	4228,67	40,47	698,5
661	4223,62	4396,72	-4	4226,31	21,05	699,45
662	4232,49	4380,05	-11	4237,41	38,77	700,17
663	4243,82	4452,21	5	4244,15	1,38	701,51
664	4249,26	4420,61	-2	4258,02	71,32	702,4
665	4252,78	4440,94	0	4253,28	1,89	703,5
666	4253,12	4385,86	-13	4261,08	62,46	703,73
667	4255,05	4457,13	4	4256,76	6,23	705,52
668	4267,87	4456,34	2	4268,09	1,07	706,51
669	4273,8	4470,22	3	4274,1	1,67	707,51
670	4276,37	4474,85	3	4280,4	29,61	708,56
671	4281,25	4448,51	-4	4287,33	47,63	709,37
672	4284,35	4456,04	-3	4289,62	39,96	710,42
673	4293,09	4428,99	-9	4301,08	61,77	710,98
674	4294,8	4487,45	4	4295,1	1,63	712,51
675	4299,06	4476,31	-1	4304,78	42,69	713,48

676	4312,74	4481,61	-1	4320,89	69,09	714,46
677	4313,35	4483,72	-1	4316,59	23,39	715,49
678	4315,65	4504,05	2	4316,28	3,04	716,51
679	4316,49	4487,29	-2	4319,42	21,56	717,47
680	4327,14	4514,68	3	4330,49	22,68	718,55
681	4333,41	4500,33	-2	4342,94	77,96	719,4
682	4335,5	4528,07	5	4336,47	3,56	720,52
683	4342,97	4526,02	2	4343,09	0,29	721,5
684	4343,33	4508,94	-3	4355,82	103,35	722,29
685	4345,06	4518,94	-1	4355,71	85,97	723,45
686	4355,65	4560,4	6	4360,15	36,97	724,63
687	4356,83	4546,09	4	4357,1	1,33	725,51
688	4363,03	4530,89	1	4374,67	93,3	726,57
689	4365,77	4527,57	-3	4376,2	89,21	727,31
690	4372,93	4579,75	5	4373,05	0,5	728,51
691	4379,25	4513,45	-9	4394,85	130,45	728,52
692	4386,87	4528,96	-4	4400,76	120,74	730,13
693	4391,25	4545,33	-3	4407,26	134,71	731,2
694	4393,48	4562,08	-1	4404,6	93,84	732,44
695	4393,96	4591,81	3	4394,1	0,36	733,5
696	4397,68	4600,91	5	4398,3	1,92	734,51
697	4410,43	4594,51	2	4413,44	22,03	735,53
698	4412,9	4569,37	-4	4433,26	178,68	735,99
699	4420,24	4583,98	-3	4440,62	173,22	737,14
700	4421,34	4622,38	7	4425,25	28,42	738,63

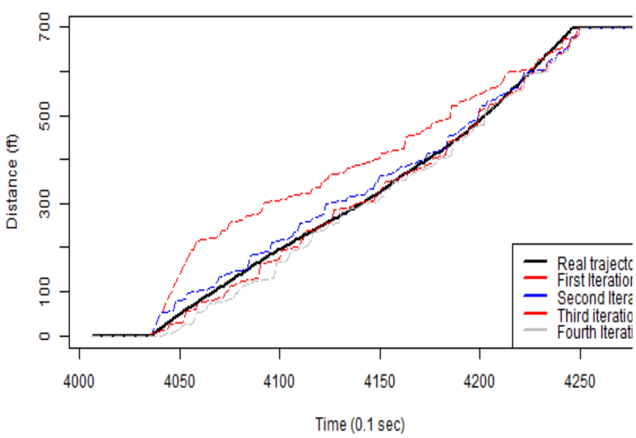
B.2 Graphics data set 2



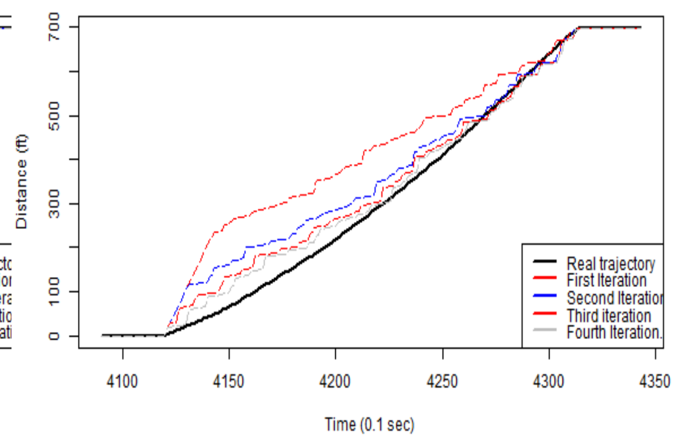
(a) Trajectory estimation veh.605



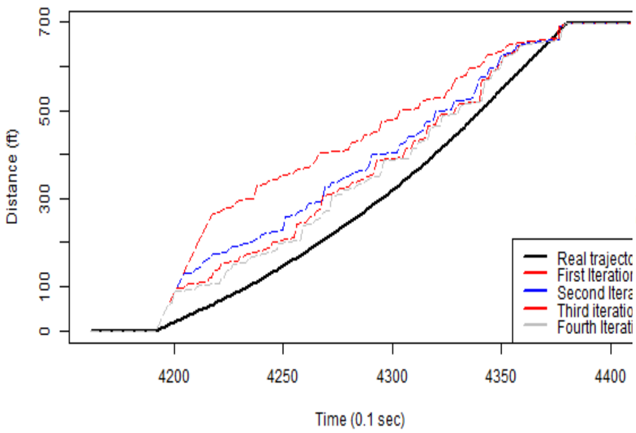
(b) Trajectory estimation veh.618



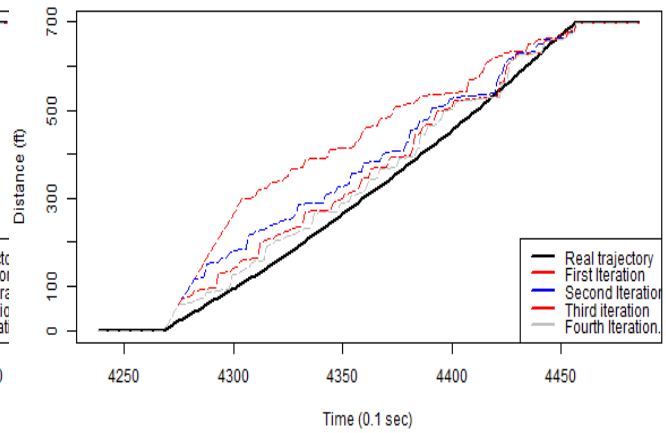
(a) Trajectory estimation veh.625



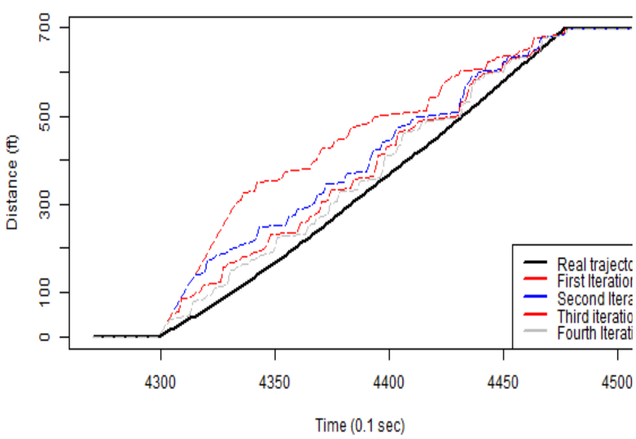
(b) Trajectory estimation veh.639



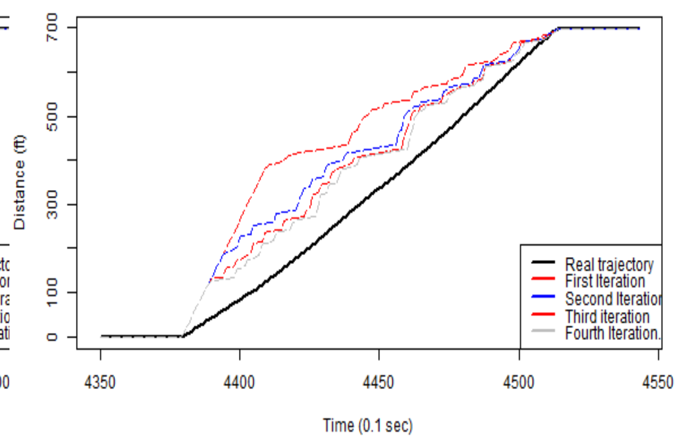
(a) Trajectory estimation veh.652



(b) Trajectory estimation veh.668



(a) Trajectory estimation veh.675



(b) Trajectory estimation veh.691

C The 701-800 interval from the data set 3

Finally, we repeat the process done before for the data set 3.

C.1 Estimated trajectory data set 3

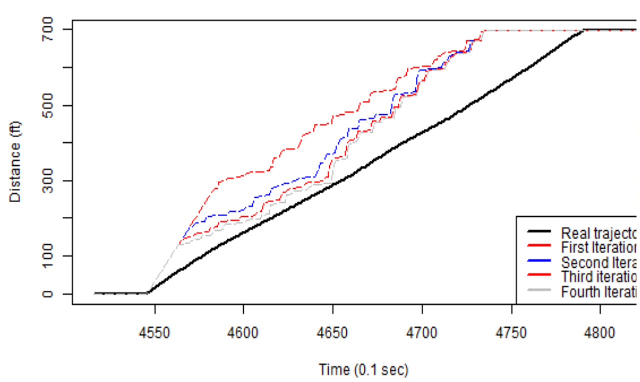
ID	**Entry Time**	**Exit time**	**Overtakes**	**e(i)**	**X(e(i))**	**Theta(e(i))**
ID	r	s	overtakes	e	xe	overvalue
701	4524,68	4793,39	15	4525,27	2,18	739,07
702	4528,84	4768,61	4	4529,09	1,29	740,05
703	4535,2	4753,27	-2	4554,32	161,46	740,88
704	4536,98	4726,17	-9	4555,18	159,59	741,19
705	4539,42	4819,59	15	4540,47	4,02	743,08
706	4543,44	4727	-10	4560,71	147,15	743,15
707	4545,7	4790,08	6	4546,28	2,25	745,06
708	4552,16	4779,93	3	4564,32	100,22	746,21
709	4555,03	4756,99	-4	4570,21	125	746,76
710	4559,73	4801,84	7	4560,28	1,79	748,06
711	4568,17	4768,81	-4	4583,58	128,4	748,76
712	4572,76	4777,01	-3	4587,4	126,87	749,84
713	4574,31	4753,95	-10	4589,32	125,51	750,23
714	4575,09	4771,91	-6	4587,8	103,86	751,69
715	4581,18	4806,93	3	4584,65	23,16	753,09
716	4586,26	4777,34	-6	4598,78	103,69	753,68
717	4588,92	4842,62	9	4589,09	0,62	755,05
718	4593,15	4838,54	6	4596,43	20,74	756,12
719	4600,26	4790,87	-4	4612,13	98,58	756,81
720	4600,94	4789,82	-8	4601,07	0,53	758,05
721	4602,83	4857,28	9	4603,09	1,41	759,06
722	4605,22	4810,95	-3	4613,08	62,97	759,94
723	4612,22	4830,04	-1	4620,81	68,93	761,02
724	4615,38	4864,69	7	4618,55	22,78	762,12

725	4615,77	4790,28	-11	4619,33	29,13	762,85
726	4617,82	4905,41	15	4618,16	1,47	764,06
727	4619,46	4834,89	-4	4623,24	28,03	764,99
728	4626,74	4845,76	-1	4628,36	11,45	766,05
729	4630,06	4944,26	19	4631,83	7,29	767,11
730	4631,98	4917,07	13	4632,05	0,11	768,05
731	4636,65	4822,33	-10	4638,73	14,86	768,98
732	4639,46	4855,31	-3	4640,55	4,92	770,04
733	4649,64	4880,61	1	4650,42	3,13	771,05
734	4651,59	4886,08	2	4652,44	3,53	772,05
735	4652,93	4867,97	-3	4653,08	0,62	773,05
736	4657,94	4977,86	19	4658,06	0,37	774,06
737	4663,42	4840,99	-12	4665,06	0,6	774,94
738	4666,32	4883,04	-3	4670,08	0,26	776
739	4667,17	4892,2	-1	4679,72	6,39	777
740	4667,49	4854,99	-12	4684,06	35,53	776,99
741	4680,07	4995,8	16	4690,12	18,83	779,55
742	4682,06	4900,63	-2	4691,82	16,1	779,97
743	4682,55	4878,9	-10	4693,58	13,55	780,52
744	4686,01	4887,97	-7	4693,07	0,77	781,81
745	4689,67	5007,15	16	4710,13	19,17	784,08
746	4692,2	4926,47	0	4710,27	19,68	784,05
747	4702,81	4921,13	-3	4709,85	6,9	784,97
748	4705,92	4899,39	-9	4711,21	1,45	785,82
749	4707,35	5072,53	25	4722,06	0,09	788,05
750	4714,45	5052,99	19	4729,17	1,45	788,87
751	4720,02	4914,85	-9	4728,07	0,6	788,68
752	4723,94	5091,92	26	4756,77	59,9	792,32
753	4724,07	4945,63	-4	4737,07	0,09	790,82
754	4726,17	4924,82	-9	4743,07	0,09	791,29
755	4726,58	4962,03	-4	4756,52	57,85	792,55
756	4742,3	4943,5	-9	4768,59	127,89	792,9

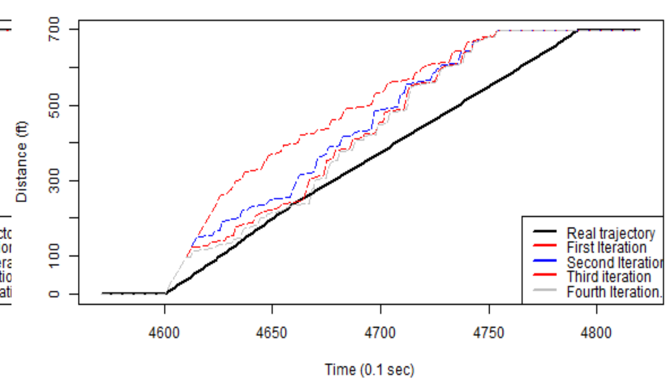
757	4742,43	4970,67	-4	4756,87	16,1	794,81
758	4742,74	5111,69	23	4762,06	0,6	797,25
759	4747,68	4978,96	-3	4762,59	22,58	796,87
760	4748,29	4948,82	-10	4765,9	25,14	797,22
761	4760,55	5001,17	-2	4768,43	12,18	798,99
762	4765,93	4971,38	-8	4769,22	2,47	799,93
763	4766,55	5137,25	24	4788,21	63,31	802,44
764	4768,87	4970,28	-12	4786,42	73,53	801,03
765	4770,65	5007,82	-3	4791,17	62,46	802,79
766	4779,32	5035,89	0	4791,87	52,23	804,05
767	4784,28	5159,73	24	4789,55	14,23	805,35
768	4786,28	4996,88	-10	4793,18	19	805,73
769	4787,2	5023,44	-5	4792,09	0,6	806,95
770	4790,5	5001,43	-10	4799,19	28,2	807,65
771	4801,62	5068,53	2	4802,29	3,27	809,05
772	4803,54	5009,93	-9	4806,69	14,23	809,94
773	4804,21	5029,3	-8	4806,57	5,54	810,99
774	4814,49	5188,47	25	4815,36	3,32	812,09
775	4818,42	5060,38	-5	4822,71	23,6	812,98
776	4818,87	5099,78	4	4821,54	13,72	814,08
777	4822,8	5038,51	-10	4823,24	2,1	815,04
778	4828,73	5136,08	8	4829,29	2,1	816,06
779	4833,46	5049,99	-11	4834,72	5,9	817,02
780	4837,61	5119,57	3	4838,05	0,94	818,06
781	4838,04	5087,88	-5	4852,59	57,51	818,77
782	4843,6	5063,82	-10	4852,74	50,02	819,67
783	4849,49	5062,84	-12	4854,71	42,01	820,8
784	4849,52	5213,99	19	4850,07	0,43	822,08
785	4850,67	5137,28	3	4859,94	26,33	823,14
786	4862,59	5112,62	-4	4873,34	91,6	823,88
787	4863,4	5091,9	-10	4865,61	14,6	824,98
788	4869,91	5234,56	20	4870,07	0,59	826,06

789	4874,3	5160,46	3	4875,59	5,07	827,06
790	4891,3	5186,07	7	4906,56	127,83	828,4
791	4891,8	5127,17	-6	4903,27	99,81	828,77
792	4891,99	5120,19	-8	4903,15	97,52	829,67
793	4896,66	5078,03	-18	4916,62	172,2	829,13
794	4903,28	5141,09	-5	4918,23	125,68	831,74
795	4906,38	5095,84	-16	4925,53	163,51	831,48
796	4912,92	5211,63	6	4913,05	0,68	834,05
797	4917,49	5164,01	-3	4934,09	142,04	834,85
798	4920,7	5183,98	-2	4940,14	167,77	835,9
799	4927,59	5230,87	8	4928,4	3,52	837,06
800	4928,31	5150,68	-10	4956,15	238,66	836,81

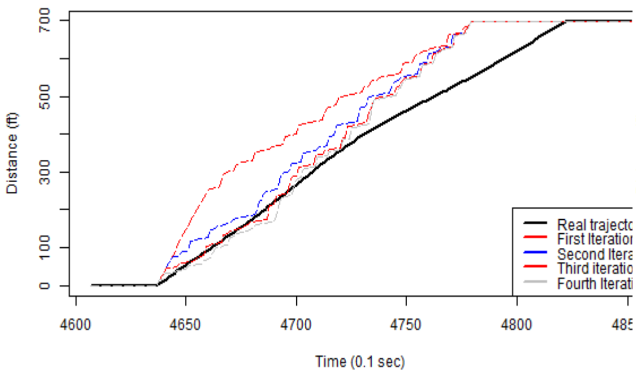
C.2 Graphics data set 3



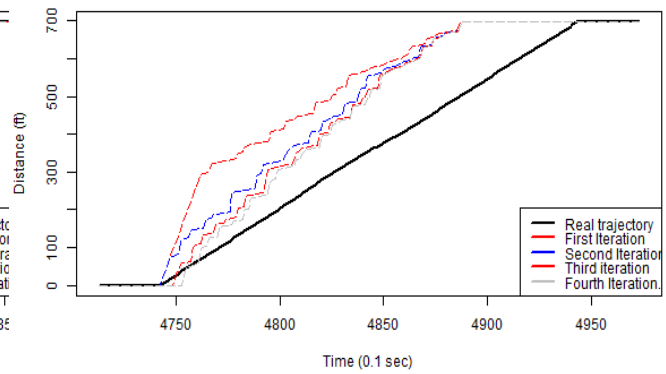
(a) Trajectory estimation veh.707



(b) Trajectory estimation veh.719



(a) Trajectory estimation veh.731



(b) Trajectory estimation veh.756

D R code

```r
setwd("C:/Users/adria_000/Desktop/ADRI/UNI/TRANSP/R trajectories
rey/ET4/") #change de working directory
source('Functions.R') #load the functions file

#####Read the NGSIM data and extract the information we want(data
treatment)####################
traj3<-read.table("trajectories-0750am-0805am.txt",) #read the file,
add options at the end
colnames(traj3)<-
c("ID","Frame","Tot_frame","G_time","L_x","L_y","G_x","G_y","L","W","C
lass","V","Acc","Lane","P_veh","F_veh","Spacing","Headway")
sum.result<-sumNGSIM(traj3)


#INITAL GUESS and first values#########
Wf=22 #20
Kf=170 #156.5
n0=38.17
iteranum=4 #number of iterations
part=5#100 partitions
results=matrix(nrow=iteranum,ncol=5)
results[,1]=seq(1:iteranum)
results=as.data.frame(results)
colnames(results)<-c("Iteration", "Error","n0","W","K")
#res.tot=matrix(nrow=r,ncol=5)
#res.tot=as.data.frame(res.tot)
#colnames(res.tot)<-c("Iteration", "Error","n0","W","K")

nn=100
l1=701
l2=800


for (r in 1:part){

for (k in 1:iteranum){
  if (k==1){
    Wf=25 #20
    Kf=170 #156.5
    n0=38.17
  }
  results[k,3]=n0
  results[k,4]=Wf
  results[k,5]=Kf

  ###############################Initial Data#################
  all.veh3<-sum.result[[1]]
  #cumulative flow at upstream
  ffun<-sum.result[[2]]
  #cumulative flow at downstream
  gfun<-sum.result[[3]]
  #inverse cumulative flow at upstream
  invffun<-sum.result[[4]]
  #inverse cumulative flow at downstream
  invgfun<-sum.result[[5]]
  #initial number of vehicles
  n0<-sum.result[[6]]
  #length of the road segment
  l<-sum.result[[7]]
  start_y<-sum.result[[8]]
  end_y<-sum.result[[9]]
```

```
  ########################Dimensions of the discretized
segment##############

  a<-floor(l)+1 #distance
  b<-ceiling(max(all.veh3$Leave)) #time value of the last vehicle
passing trough g, each step is 1/10 seconds
  n=1698 #total vehicle number



  ############Data analysis for each vehicle
(av.speed,st.dev,max,min)#####

  all.veh3=all.veh3[order(all.veh3[,2]),]#order for entrance
  all.veh3$Entrypos=seq(1:n)#number 1 to n in entrance order
  all.veh3=all.veh3[order(all.veh3[,3]),]
  all.veh3$Exitpos=seq(1:n)
  all.veh3$overtakes=all.veh3$Exitpos-all.veh3$Entrypos #Mi
  all.veh3$Ai=all.veh3$overtakes/(all.veh3$Leave-all.veh3$Enter) #Ai
  all.veh3=all.veh3[order(all.veh3[,2]),]#order for entrance for doing
the estimation


  ###############Trajectory estimation of n vehicles################

  #section 1-100 FIFO VIOLATION

  all.traj.nofifo3.beta=matrix(nrow=b-1,ncol=n+1)
  all.traj.nofifo3.beta[,1]=seq(1:(b-1))
  overfun=matrix(nrow=b-1,ncol=n)
  pb<-txtProgressBar(max=n,style=3)
  for (p in l1:l2){
    traj.result=matrix(nrow=b-1,ncol=4)
    traj.result[,1]=all.traj.nofifo3.beta[,1]
    entrytime=all.veh3[p,2]
    exittime=all.veh3[p,3]
    p1=ffun(entrytime)
    theta1=all.veh3$overtakes[p]/(exittime-entrytime)
    theta2=(p1*exittime-
entrytime*(p1+all.veh3$overtakes[p]))/(exittime-entrytime)
    for (i in 1:(b-1)){ #is starting at point 1200
      theta=theta1*i+theta2
      traj.result[i,2]<-(60*5280/36000)*(i-invffun(theta))
      if(traj.result[i,2]<0){traj.result[i,2]<-0
      }else if(traj.result[i,2]>l){traj.result[i,2]<-l
      }
      traj.result[i,3]<-bisection(gfun,i,l,theta+n0,0.1,20)
      traj.result[i,4]<-min(traj.result[i,2],traj.result[i,3])
      overfun[i,p]=theta+n0
    }

    all.traj.nofifo3.beta[,p+1]=traj.result[,4]
    setTxtProgressBar(pb,p)
  }
  all.traj.nofifo3.beta<-as.data.frame(all.traj.nofifo3.beta)

  ####plot###
  f=756
  if(k==1){
  entrytime=all.veh3[f,2]
  exittime=all.veh3[f,3]
```

```
  traj.single=real.traj3[,c(1,f+1)]  #Real trajectory
  traj.single=as.data.frame(traj.single)
  colnames(traj.single)<-c("Frame", "L_y")
  traj.single=traj.single[traj.single$Frame>(entrytime-30) &
traj.single$Frame<(exittime+30),]

  traj.result=all.traj.nofifo3.beta[,c(1,f+1)]  #Estimated traj. (no
FIFO)
  traj.result=as.data.frame(traj.result)
  colnames(traj.result)<-c("Frame", "L_y")
  traj.result=traj.result[traj.result$Frame>(entrytime-30) &
traj.result$Frame<(exittime+30),]

  plot(traj.single$Frame, traj.single$L_y, type='l', lwd=2, xlab="Time
(0.1 sec)", ylab="Distance (ft)", xlim=c(entrytime-30, exittime+30))
  lines(traj.result[,1],traj.result[,2],lty=5, lwd=1.5, col=90)
  }
  if(k!=1){
  entrytime=all.veh3[f,2]
  exittime=all.veh3[f,3]
  traj.result=all.traj.nofifo3.beta[,c(1,f+1)]  #Estimated traj. (no
FIFO)
  traj.result=as.data.frame(traj.result)
  colnames(traj.result)<-c("Frame", "L_y")
  traj.result=traj.result[traj.result$Frame>(entrytime-30) &
traj.result$Frame<(exittime+30),]
  col=k*6
  lines(traj.result[,1],traj.result[,2],lty=5, lwd=1.5, col=col)
  }

  #Estimation

  ########################Parameter Estimation
  #First we prepare the all.veh matrix and other inputs


  all.veh3=all.veh3[,c(1,2,3,6)]
  all.veh3=all.veh3[order(all.veh3[,2]),]#order for entrance for doing
the estimation
  colnames(all.veh3)<-c("ID","r","s","overtakes")
  pb<-txtProgressBar(max=n,style=3)
  all.veh3$e=NA
  all.veh3$xe=NA
  all.veh3$overvalue=NA
  for (p in l1:l2){ #last ones are not reaching 698meters
    all.veh3$e[p]=bisec2(p,60,0.1,20) #watch out the p and the
velocity
    all.veh3$xe[p]=(all.traj.nofifo3.beta[all.veh3$e[p],p+1])
    all.veh3$overvalue[p]=overfun[all.veh3$e[p],p]
    setTxtProgressBar(pb,p)
  }


  #defining the parameteres for the next iteration
  all.veh3par=all.veh3[c(l1:l2),]
  par=para.estimation(ffun,gfun,all.veh3par,cmr,100)
  n0=colMeans(par)[[1]]
  Wf=colMeans(par)[[2]]
  Kf=colMeans(par)[[3]]/5
```

```
  #Error
  #l2-l1=100
  errors3.bis=matrix(nrow=100,ncol=3)
  errors3.bis[,1]=seq(1:100)
  errors3.bis=as.data.frame(errors3.bis)
  pb<-txtProgressBar(max=n,style=3)
  t=1
  for(p in (l1:l2)){  #select a vehicle
    entrytime=all.veh3[p,2]
    exittime=all.veh3[p,3]

    traj.single=real.traj3[,c(1,p+1)]  #Real trajectory
    traj.single=as.data.frame(traj.single)
    colnames(traj.single)<-c("Frame", "L_y")
    traj.single=traj.single[traj.single$Frame>(entrytime) &
traj.single$Frame<(exittime+40),]

    traj.result=all.traj.nofifo3.beta[,c(1,p+1)]  #Estimated traj. (no
FIFO)
    traj.result=as.data.frame(traj.result)
    colnames(traj.result)<-c("Frame", "L_y")
    traj.result=traj.result[traj.result$Frame>(entrytime) &
traj.result$Frame<(exittime+40),]

    er=seq(from=traj.result$Frame[1],
to=traj.result$Frame[length(traj.result$Frame)], by=1)
    er<-as.data.frame(er)
    for(i in 1:length(er[,1])){
      c=traj.result$L_y[traj.result$Frame==er[i,1]]
      d=traj.single$L_y[traj.single$Frame==er[i,1]]
      er[i,2]=abs(c-d)
      er[i,3]=d
    }
    colnames(er)<-c("Frame","Error","Real")
    mp=sum(er$Error)/sum(er$Real)
    errors3.bis[t,2]=mp
    t=t+1
    setTxtProgressBar(pb,p)
  }
  meanit=colMeans(errors3.bis)[[2]]
  results[k,2]=meanit
  if (k!=1){
    tolw=abs(results[k-1,4]-results[k,4])/results[k-1,4]
    tolk=abs(results[k-1,5]-results[k,5])/results[k-1,5]
    if (tolw<= 0.002 & tolk<=0.007){
      break
    }

  }

}
####Extracting the plot######
legend("bottomright",c("Real trajectory","First Iteration","Second
Iteration","Third iteration","Fourth
Iteration..."),col=c(1,90,12,18,24),lty=c(1,1),lwd=c(2,2))
str = sprintf('plot%i.png', f)
dev.copy(png,str,width=607,height=356)
dev.off()

#mp=which.min(results[,2])#### this is to compute the minimum
```

```
#res.tot[r,]=results[mp,]

res.tot[r,]=results[k,]


l1=l1+100
l2=l2+100
}

#extracting dataaa
all.veh3par[,-1] <-round(all.veh3par[,-1],2) #rounding to 2 decimals
write.table(all.veh3par,"C:/Users/adria_000/Desktop/ADRI/UNI/TRANSP/R
trajectories rey/ET4/dataset2.xls",dec=",")
```

```
summary.veh<-function(traj, start, end){
  result<-approx(traj$L_y,traj$Frame, xout=c(start,end)) #lineal
interpolation--return interpolated y at points(xout) start and end
  return(result[[2]]) #only take the 2 column---ri and si(entry time
and exit time)
}

sumNGSIM<-function(traj3){
  start_y<-578 #we define the start and the end
  end_y<-578+698
  l<-end_y-start_y
  nlanes=length(unique(traj3$Lane))
  traj3<-traj3[,c(1,2,6)] #take only id, frame and local postion
  n=length(unique(traj3$ID)) #count the diferent ID(vehicles) there
are
  pb<-txtProgressBar(max=n,style=3)#progressbar for follow the
process, pb is a object of this classrep()
  i=1
  t.travel=matrix(rep(-1,3*n),ncol=3) #matrix nx3 with -1 values
  for (id in unique(traj3$ID)){ #go trhough uniques ID
    t.travel[i,]<-c(id,summary.veh(traj3[traj3$ID==id,],start_y,
end_y)) #Introduce to the i row: id,ri,si
    i=i+1
    setTxtProgressBar(pb,i)#progressbar of each iteration
  }
  t.travel<-as.data.frame(t.travel); #returns a data frame
  colnames(t.travel)<-c("ID","Enter","Leave") #defining the colnames
of the data frame above
  #construct cumulative flow at entrance/exit
  f.raw<-approxfun(sort(t.travel$Enter),
seq(0:(sum(!is.na(t.travel$Enter))-1)),rule=2)#aprroxfun F(t) from
vehicles and his enter time
  g.raw<-approxfun(sort(t.travel$Leave),
seq(0:(sum(!is.na(t.travel$Leave))-1)),rule=2) #x is the time and y is
the sum of non-zero entries
  invf.raw<-approxfun(seq(0:(sum(!is.na(t.travel$Enter))-1)),
sort(t.travel$Enter), rule=2) #inverse function F^-1(t)
  invg.raw<-approxfun(seq(0:(sum(!is.na(t.travel$Leave))-1)),
sort(t.travel$Leave), rule=2)

  #initial number of vehicles (estimation starts at t=120s, the warm-
up period)

  #observed cumulative flow at entrance/exit
  ffun<-function(t){return(ifelse(t<1200,0,f.raw(t)-
f.raw(1200)))}#ffun begins at t1=1200, t=0 beyond this point
  gfun<-function(t){return(ifelse(t<1200,0,g.raw(t)-g.raw(1200)))}
  invffun<-function(n){return(invf.raw(n+f.raw(1200)))}
  invgfun<-function(n){return(invg.raw(n+g.raw(1200)))}
  t.travel<-t.travel[t.travel$Enter>1200 &
!is.na(rowSums(t.travel)),]#check there is no NA and update t.travel
to t>1200
  return(list(t.travel,ffun,gfun,invffun,invgfun,n0,l,start_y,end_y))
}

form<-function(gfun, t, l, p, x){
  return(gfun(t-(l-x)/(Wf*5280/36000))+(Kf*5/5280)*(l-x)-p)
}

bisection<-function(gfun, t, l, p, tol, max_iter){
  x1<-0
```

```
  x2<-l
  if (sign(form(gfun,t,l,p,x1))==sign(form(gfun,t,l,p,x2))){
    ifelse (sign(form(gfun,t,l,p,x1))>0, return(l),
return(0))#defining the limits over means l, less means 0
  }
  else{ #doing iterations in the bisection method
    it<-1
    while(it<=max_iter){
      x3<-(x1+x2)/2
      if(form(gfun,t,l,p,x3)==0 | (x2-x1)/2<tol){return(x3)}
      it<-it+1
      ifelse (sign(form(gfun,t,l,p,x3))==sign(form(gfun,t,l,p,x1)),
x1<-x3, x2<-x3)
    }
    return(x3)
  }
}

traject3.veh<-function(p,t){
  entrytime=all.veh3[p,2]
  exittime=all.veh3[p,3]
  traj.result=all.traj.nofifo3.beta[,c(1,p+1)]
  traj.result=as.data.frame(traj.result)
  colnames(traj.result)<-c("Frame", "L_y")
  return(traj.result[t,2])

}
fore<-function(p,v,t){
  traj.result=all.traj.nofifo3.beta[,c(1,p+1)]
  traj.result=as.data.frame(traj.result)
  colnames(traj.result)<-c("Frame", "L_y")
  frameini<-head(which(traj.result$L_y!=0),n=1L)
  v=v*(5280)/36000 #to ft/(1/10segon)
  return(traject3.veh(p,t)-v*(t-frameini))
}
bisec2<-function(p,v,tol,max_iter){
  traj.result=all.traj.nofifo3.beta[,c(1,p+1)]
  traj.result=as.data.frame(traj.result)
  colnames(traj.result)<-c("Frame", "L_y")
  frameini<-head(which(traj.result$L_y!=0),n=1L)
  framefini<-head(which(traj.result$L_y==698),n=1L)
  if(sign(fore(p,v,frameini))==sign(fore(p,v,framefini))){
    ifelse
(sign(fore(p,v,frameini))>0,return(framefini),return(frameini))
  }
  else{ #doing iterations in the bisection method
    it<-1
    while(it<=max_iter){
      frame3<-(frameini+framefini)/2
      if(fore(p,v,frame3)==0 | (framefini-
frameini)/2<tol){return(frame3)}
      it<-it+1
      ifelse (sign(fore(p,v,frame3))==sign(fore(p,v,frameini)),
frameini<-frame3, framefini<-frame3)
    }
    return(frame3)
    }
}
#####ESTIMATION FUNCTION############
para.estimation<-function(ffun, gfun, all.veh, cmr, runs, trace=F){
  #gfun()-downstream cumulative flow function
```

```
  #ffun()-upstream cumulative flow function
  #all.veh-a data.frame of vehicle trajectory information with four
columns
  #all.veh$ID-vehicle id
  #all.veh$r-vehicle entrance time (r_i) in 1/10 second
  #all.veh$s-vehicle exit time (s_i) in 1/10 second
  #all.veh$e-vehicle transition time (e_i) in 1/10 second
  #all.veh$xe-position at time e
  #all.veh$overvalue- function theta at time e
  #all.veh$overtakes-value of M_i
  #mp.rate-market penetration rate
  #cmr-correct matching rate---error reidentification match
  #runs-number of runs for each setting
  Ks<-NULL;Ws<-NULL;n0s<-NULL
  for (i in 1:runs){
    #sample from the trajectory data according to the correct matching
rate
    t.table=all.veh[base::sample(1:nrow(all.veh), nrow(all.veh)*cmr),]
#from the sample 1 to n, take cmr*n
    #estimate initial number of vehicles
    n0.hat<-mean(gfun(t.table$s)-ffun(t.table$r)+t.table$overtakes)
    #solve the nonlinear least square estimation problem using the nls
function, see details (help nls)
    fit<-nls(overvalue~gfun(e-(l-xe)/(w1*5280/3600/10))+kj1/5280*(l-
xe),data=t.table, start=list(w1=20,
kj1=700),nls.control(maxiter=18000,minFactor=1e-
17,tol=0.18),trace=trace)
    Ws<-c(Ws,unname(coef(fit)[1]))
    Ks<-c(Ks,unname(coef(fit)[2]))
    n0s<-c(n0s, n0.hat)
  }
  par<-data.frame(n0s, Ws, Ks)
  par=as.data.frame(par)

  return(par)
}


traj.singlecar<-function(car){
  entrytime=all.veh3[car,2]
  exittime=all.veh3[car,3]

  traj.single=real.traj3[,c(1,car+1)]  #Real trajectory
  traj.single=as.data.frame(traj.single)
  colnames(traj.single)<-c("Frame", "L_y")
  traj.single=traj.single[traj.single$Frame>(3900) &
traj.single$Frame<(4650),]
  lines(traj.single[,1],traj.single[,2],col="green")

  return(traj.single)
  }


#pol_over<-function(p,i)
  #mi=all.veh3$overtakes[p]
  #ri=all.veh3[p,2]
  #si=all.veh3[p,3]

  #lloc=head(which(av_data$i==p),n=1L)
  #nj=av_data[lloc,3]
  #tk=av_data[lloc,1]
```